# APPENDIX B. MODEL DOCUMENTATION

## Table of Contents



## Tables

# 1 Model and Rationale for Conversion to Visual Basic for Applications

The primary goal of mechanistic modeling for the Lower Passaic River Study Area (LPRSA) remedial investigation/feasibility study (RI/FS) is to develop a predictive relationship among chemical concentrations in sediment, water, and tissue. This relationship will be used to derive preliminary remediation goals (PRGs) in sediment and compare remedial alternatives for chemicals that are present in fish tissue, water, and sediment at concentrations associated with unacceptable risk. The mechanistic model is based on algorithms and equations initially established by Gobas (1993). This model has been used as the basis for many subsequent updates and iterations of Gobas-type models, including refinements and simplifications (Arnot and Gobas 2004; Morrison et al. 1996, 1997). The driving force of these fugacity-based models is phase partitioning. The first type of partitioning occurs between water and the organism, and the second occurs during the digestion process between prey items or ingested sediment and the organism.

Mechanistic bioaccumulation models have been used in a broad range of environments (i.e., lakes, rivers, and estuaries), as described in Section 2.1 of the main text and in Appendix A. This LPRSA bioaccumulation model was adapted from the Arnot-Gobas (2004) model but was transferred into Visual Basic for Applications® (VBA) code.

Passing inputs and outputs for the VBA version of the mechanistic model is accomplished through the use of Microsoft Excel® spreadsheets. An effort was made to avoid complicated (although perhaps more efficient) coding in order to preserve the transparency of the way the model functions. Use of Excel® for the biotic model interface facilitates the concurrent use of Monte Carlo software for enhanced uncertainty and sensitivity analyses. This combination of software makes it possible to run multiple iterations of the model. This appendix describes the components of the model and presents the VBA code used to run the model. The acronyms provided in the model and sub-model explanations (inputs and outputs) are the same as those used in the VBA code, unless otherwise indicated.

The remainder of this appendix is organized as follows:

- Section 2 – General Processes and Structure of the Model
- Section 3 – Complete VBA Model Code
- Section 4 – References

# 2 General Processes and Structure of the Model

This section presents an overview of the species and modeling areas used in the model, modeling setup and assumptions, equations used to represent physical and chemical processes, equations used to represent general biological processes, and an overview of species-specific calculations.

## 2.1 Species and Modeling Areas

The species groups (and representative species, where applicable) that were modeled for the LPRSA bioaccumulation model are as follows:

- Phytoplankton/algae
- Zooplankton
- Benthic invertebrate deposit feeders (DEPs)
- Benthic invertebrate detritivores (DETs)
- Benthic invertebrate carnivore/omnivores (C/Os)
- Small filter feeding fish
- Small forage fish
- Blue crab
- Benthic omnivorous fish (represented by carp)
- Benthic omnivorous/invertivorous fish (represented by catfish)
- Small invertivorous fish (represented by white perch)
- Piscivorous fish (represented by American eel)
- Piscivorous freshwater fish (represented by freshwater bass)

Additional information about the model structure, selected trophic levels, and representative species are provided in the main text and in Appendices E and F.

For each species, a modeling area was determined based on site-specific catch information, as well as on literature information regarding the potential habitat of the various species (e.g., information regarding species salinity tolerance). This information is discussed in detail in Section 3.2.3 of the main report. The modeling areas used for each species are as follows:

- **RM 0 to RM 17.4 (i.e., site wide)** – The site-wide modeling area was selected for small filter-feeding fish, small forage fish (mudflats only), blue crab, white perch, and American eel.
- **RM 4 to RM 17.4** – This modeling area was selected for catfish.
- **RM 7 to RM 17.4** – This modeling area was selected for carp and bass.

Additionally, it should be noted that for small forage fish, the modeling area was defined to include only mudflat areas within the LPRSA, which were defined as shallow areas with gradual river bottom slopes (corresponding to lower water velocities). This definition was based on information regarding the inability of these fish to tolerate higher water velocities and their preference for shallow water habitats; catch data from the LRPSA for these species corroborated this modeling area definition (see Section 3.2.3 of the main report for more information).

This definition of the modeling area for small forage fish necessitated separate code specific to the mudflat areas to estimate exposure for small forage fish and their prey (i.e., in addition to the river-wide calculations for other species). This code can be seen in the complete model code presented in Section 3 of this appendix.

For catfish, carp, and bass (i.e., species for which the modeling area was identified as smaller than the site due to their reduced salinity tolerance), separate model code was not needed. Rather, controls in the model spreadsheet were used to determine the chemical concentrations to which these species were exposed.

## 2.2 Model Setup and Assumptions

The LPRSA bioaccumulation model was designed around the premise that a single equation may be used to represent the exchange of non-ionic organic chemicals between an organism and its environment (Arnot and Gobas 2004). The conceptual equation, which underlies the model and describes the net flux of a parent chemical being absorbed or deposited ($dM_B$) by an organism at any time (dt), is:

$$\frac{dM_B}{dt} = \left\{ W_B \cdot \left( k_1 \cdot [m_O \cdot C_{WD,O} + m_P C_{WD,P}] + k_D \cdot \sum_i (P_i \cdot C_{D,i}) \right) \right\} - (k_2 + k_E + k_M) \cdot M_B$$

**Equation 1**

Where:

| | | |
|---|---|---|
| $dM_B$ | = | mass of chemical in organism (g), at a time t |
| dt | = | point in time |
| $W_B$ | = | wet weight of organism (kg) |
| $k_1$ | = | clearance rate constant for water ventilated by organism (L/kg×day) |
| $m_o$ | = | fraction of respiratory ventilation involving overlying water (unitless) |
| $m_p$ | = | fraction of respiratory ventilation involving sediment-associated porewater (unitless) |
| $C_{WD,O}$ | = | total freely dissolved chemical concentration in overlying water (g/L) |
| $C_{WD,P}$ | = | total freely dissolved chemical concentration in sediment-associated porewater (g/L) |
| $k_D$ | = | clearance rate constant via ingestion of food and water (kg/kg×day) |
| $P_i$ | = | fraction of the diet composed of prey item i (unitless) |
| $C_{D,i}$ | = | chemical concentration in prey item i (g/kg) |
| $k_2$ | = | gill and skin elimination rate constant (1/day) |
| $k_E$ | = | rate constant for chemical elimination via excretion into egested feces (1/day) |
| $k_M$ | = | metabolic biotransformation rate constant of the chemical (1/day) |
| $M_B$ | = | mass of chemical in organism (g) |

For model calibration, a steady state version of the model is required. The steady state solution is derived by first dividing both sides of Equation 1 by $W_B$ to convert mass into a wet weight concentration of the chemical in the organism ($C_B$), then setting the rate of change term $dC_B/dt$ to zero (i.e., no change = steady state). So, Equation 2 is the steady state equation used to assess biomagnification and bioaccumulation up the food chain:

$$C_B = \frac{k_1 \times (m_O \times C_{WD,O} + m_P \times C_{WD,P}) + k_D \times \sum P_i \times C_{D,i}}{k_2 + k_E + k_G + k_M}$$

**Equation 2**

Where:

| | | |
|---|---|---|
| $C_B$ | = | chemical concentration in organism (g/kg ww) |
| $k_1$ | = | clearance rate constant for water ventilated by organism (L/kg×day) |
| $m_o$ | = | fraction of respiratory ventilation involving overlying water (unitless) |
| $C_{WD,O}$ | = | total bioavailable chemical concentration in overlying water (g/L) |
| $m_p$ | = | fraction of respiratory ventilation involving sediment-associated porewater (unitless) |
| $C_{WD,P}$ | = | total freely dissolved chemical concentration in sediment-associated porewater (g/L) |
| $k_D$ | = | clearance rate constant via ingestion of food and water (kg/kg×day) |
| $P_i$ | = | fraction of the diet composed of prey item i (unitless) |
| $C_{D,i}$ | = | chemical concentration in prey item i (g/kg) |
| $k_2$ | = | gill and skin elimination rate constant (1/day) |
| $k_E$ | = | rate constant for chemical elimination via excretion into egested feces (1/day) |
| $k_G$ | = | growth rate constant (1/day) |
| $k_M$ | = | metabolic biotransformation rate constant of the chemical (1/day) |

A number of specific sub-models are used to define the rate coefficients in the steady-state equation. These sub-models can be broken down into three categories: physical, chemical, and biological processes. Additional variables are required to parameterize the sub-models and are defined below as the sub-models are presented.

## 2.3 General Biological Processes

The general biological processes included in the model are described below. In some cases, the acronyms used by Arnot and Gobas (2004) and described below vary slightly from the acronyms used in the VBA model. For example, the clearance rate via respiration is described below as $k_1$ and is included in the model code as K1, and the dietary absorption efficiency of lipid $\varepsilon_L$ is included in the model code as eL.

### 2.3.1 Dietary apportionment

Diets of fish and invertebrates are likely to be variable because of opportunistic feeding behavior and seasonal and spatial variations in prey availability. The presence of natural fluctuations in dietary preferences is addressed by normalizing dietary fractions across a "menu" of possible food items. Each trophic group is assigned one best estimate of dietary items and portion of each dietary item. Details on selected best estimates for dietary items and portions of dietary items are presented in the main text and in Appendix H.

Dietary exposure to ingested prey tissue and ingested sediment affects the consumer during the digestion process. Phase partitioning occurs across the gut wall, and chemicals may be absorbed into the tissues or expelled from the tissues into the gut contents. This exchange of chemicals during the digestive process is discussed in greater detail in Section 2.3.3.

### 2.3.2 Direct contact through water exposure – phase partitioning

Organic chemicals are thought to partition between lipid, protein, carbohydrate (collectively known as NLOM and NLOC), and water. The sorption and storage of chemicals may occur to a certain extent in each of these media for each organism modeled. Therefore, an organism-water partitioning coefficient ($K_{BW}$), which results from direct contact with water during respiration, is determined for each organism according to Equation 3.

$$k_{BW} = \frac{k_1}{k_2} = V_{LB} \times K_{OW} + V_{NB} \times \beta \times K_{OW} + V_{WB}$$

**Equation 3**

Where:

| | | |
|---|---|---|
| $K_{BW}$ | = | organism-specific water partitioning coefficient (unitless) |
| $k_1$ | = | clearance rate constant for water ventilated by organism (L/kg×day) |
| $k_2$ | = | gill and skin elimination rate constant (1/day) |
| $V_{LB}$ | = | lipid fraction of the organism (unitless) |
| $V_{NB}$ | = | NLOM fraction of the organism (unitless) |
| $V_{WB}$ | = | water fraction of the organism (unitless) |
| $\beta$ (BETA) | = | NLOM-octanol proportionality constant (unitless) |
| $K_{OW}$ | = | chemical-specific octanol-water partition coefficient (kg/L) |

When calculating $K_{BW}$ for phytoplankton, $V_{NB}$ is replaced by the NLOC-octanol proportionality constant (GAMMA), as this constant GAMMA affects partitioning between water and NLOC (see Section 2.4.2).

In order to estimate the parameters $k_1$ and $k_2$, Arnot and Gobas relied on the following set of sub-models (Arnot and Gobas 2004).

The gill uptake rate constant ($k_1$) describes the rate at which chemicals are absorbed from water across the membranes of the gills and skin. It is considered a function of the ventilation rate ($G_v$) and the diffusion rate across the surface, as shown in Equations 4 and 5.

$$k_1 = \frac{E_W \times G_v}{W_B}$$

**Equation 4**

$$G_V = \frac{1{,}400 \times W_B^{\ 0.65}}{C_{OX}}$$

**Equation 5**

Where:

| | | |
|---|---|---|
| $k_1$ | = | clearance rate constant for water ventilated by organism (L/kg × day) |
| $E_W$ | = | chemical uptake efficiency across the gills as a percentage (%) |
| $G_v$ | = | ventilation rate (L/day) |

$W_B$ = weight of the organism (kg)
$C_{ox}$ = dissolved oxygen concentration (mg $O_2$/L)

The available DO data from the LPRSA (Windward [in prep]) were used to develop a site-specific relationship between $C_{ox}$ as a fraction of saturated dissolved oxygen concentration ($DO_{sat}$) and water temperature ($T_w$) (Equation 6).

$$\frac{C_{ox}}{DO_{sat}} = -0.24T_w + 14.04$$

**Equation 6**

Arnot and Gobas (2004) proposed a different method of calculating the gill uptake rate ($k_1$) for algae and macrophytes. Instead of the equation presented above (Equation 6), the following relationship was recommended (Equation 7).

$$k_1 = \frac{1}{\left(A + \frac{B}{K_{ow}}\right)}$$

**Equation 7**

Where:

$k_1$ = clearance rate constant for water ventilated by organism (L/kg × day)
A = resistance constants of the algae or macrophytes to the uptake of the chemical through aqueous phases (unitless)
B = resistance constants of the algae or macrophytes to the uptake of the chemical through organic phases (unitless)
$K_{OW}$ = chemical-specific octanol-water partition coefficient (kg/L)

Based on empirical data described more fully in Arnot and Gobas (2004), default values of $6.0 \times 10^{-5}$ and 5.5 were selected for constants A and B, respectively.

An additional parameter has been introduced to account for the ventilation of particulates by carp during feeding (as discussed in Section 3.3.1 of the main report). This chemical-specific parameter, called the particulate ventilation constant (CPV), accounts for increased exposure to chemicals in sediment and particulates via the gill ventilation that occurs as a result of carp feeding habits (Section 3.1.4 of the calibration report). This process is not accounted for mechanistically in the bioaccumulation model, and thus CPV acts as a multiplier on chemical intake via respiration to account for the absence of this mechanism (Equation 8).

$$\text{Ventilation Uptake} = CPV \times \left(k_1 \times \left(m_O \times C_{WD,O} + m_p \times C_{WD,P}\right)\right)$$

**Equation 8**

Where:

CPV = particulate ventilation constant (unitless)
$k_1$ = clearance rate constant for water ventilated by organism (L/kg×day)
$m_o$ = fraction of respiratory ventilation involving overlying water (unitless)
$C_{WD,O}$ = total bioavailable chemical concentration in overlying water (g/L)
$m_p$ = fraction of respiratory ventilation involving sediment-associated porewater (unitless)
$C_{WD,P}$ = total freely dissolved chemical concentration in sediment-associated porewater (g/L)

The gill elimination rate constant, $k_2$, describes the rate at which chemicals are removed from the organism across the gill membrane. Closely related to $k_1$, inasmuch as both constants are sensitive to ventilation rate and permeability across the surface of the gill membrane, $k_2$ is defined in Equation 9.

$$k_2 = \frac{k_1}{K_{BW}}$$

**Equation 9**

Where:

| | | |
|---|---|---|
| $k_2$ | = | gill and skin elimination rate constant (1/day) |
| $k_1$ | = | clearance rate constant for water ventilated by organism (L/kg × day) |
| $K_{BW}$ | = | organism-specific water partitioning coefficient (unitless) |

Because bioaccumulation is defined by the ratio of $k_1$ to $k_2$, any errors that may occur in the selection of appropriate $G_V$ and $E_W$ values will be canceled out in the model. Therefore, the model is relatively insensitive to parameterization errors in $G_V$ and $E_W$, which makes it possible to represent the ventilation rate and chemical uptake efficiency across the gill membrane with a single equation for a variety of species.

### 2.3.3 Direct contact through dietary exposure – phase partitioning

In addition to direct exposure to chemicals in the water, organisms may be exposed to chemicals present in ingested prey items.

The dietary uptake rate constant, $k_D$, defines the rate at which chemicals are removed from the gastrointestinal tract of an organism and absorbed into tissue. The dietary uptake rate constant is defined in Equation 10.

$$k_D = \frac{E_D \times G_D}{W_B}$$

**Equation 10**

Where:

| | | |
|---|---|---|
| $k_D$ | = | clearance rate constant via ingestion of food and water (kg/kg×day) |
| $E_D$ | = | dietary chemical transfer efficiency (unitless) |
| $G_D$ | = | feeding rate (kg/day) |
| $W_B$ | = | wet weight of organism (kg) |

$E_D$ has been shown to rely heavily on the $K_{OW}$ value of the chemical being absorbed, and therefore was defined by Arnot and Gobas (2004) based on a two-phase lipid-water resistance model (Equation 11).

$$E_D = (3.0 \times 10^{-7} \times K_{OW} + 2.0)^{-1}$$

**Equation 11**

The first and last terms in this equation are defined as dietary uptake constants A and B, respectively (EDA and EDB).

Feeding rates are best defined using site-specific empirical data, if such data are available. However, if such information does not exist for a particular site being modeled, feeding rate $G_D$ may be defined as the following for fish, zooplankton, and aquatic invertebrate species (Equation 12).

$$G_D = \left(0.022 \times W_B^{\ 0.85}\right) \times \exp^{(0.06 \times T)}$$

**Equation 12**

Where:

| | | |
|---|---|---|
| $G_D$ | = | feeding rate (kg /day) |
| $W_B$ | = | wet weight of organism (kg) |
| T | = | water temperature (°C) |

Chemicals may also be eliminated from an organism through fecal egestion, which is defined by the fecal elimination rate constant $k_E$ (Equation 13).

$$k_E = G_F \times E_D \times \frac{K_{GB}}{W_B}$$

**Equation 13**

Where:

| | | |
|---|---|---|
| $k_E$ | = | fecal elimination rate constant (1/day) |
| $G_F$ | = | fecal egestion rate (kg feces/kg organism x day) |
| $E_D$ | = | dietary chemical transfer efficiency (unitless) |
| $K_{GB}$ | = | Partition coefficient of the chemical between gut and organism (unitless) |
| $W_B$ | = | wet weight of organism (kg) |

The fecal egestion rate $G_F$ is a function of how digestibility of the various components of the diet (Equation 14).

$$G_F = \{[(1-\varepsilon_L) \times V_{LD}] + [(1-\varepsilon_N) \times V_{ND}] + [(1-\varepsilon_P) \times V_{PD}] + [(1-\varepsilon_W) \times V_{WD}]\} \times G_D$$

**Equation 14**

Where:

| | | |
|---|---|---|
| $G_F$ | = | fecal egestion rate (kg feces/kg organism x d) |
| $\varepsilon_L$ | = | dietary assimilation efficiencies of lipid (unitless) |
| $\varepsilon_N$ | = | dietary assimilation efficiencies of NLOM (unitless) |
| $\varepsilon_P$ | = | dietary assimilation efficiencies of NLOC (unitless) |
| $\varepsilon_W$ | = | dietary assimilation efficiencies of water (unitless) |
| $V_{LD}$ | = | lipid fraction of the diet (unitless) |
| $V_{ND}$ | = | NLOM fraction of the diet (unitless) |
| $V_{PD}$ | = | NLOC fraction of the diet (unitless) |
| $V_{WD}$ | = | water fraction of the diet (unitless) |
| $G_D$ | = | feeding rate (kg /day?) |

The partitioning coefficient between the gut contents of the organism and its tissue is estimated as shown in Equation 15.

$$K_{GB} = \frac{([V_{LG} \times K_{OW}] + [V_{NG} \times \beta] + [V_{PG} \times \gamma \times K_{OW}] + V_{WG})}{([V_{LB} \times K_{OW}] + [V_{NB} \times \beta \times K_{OW}] + V_{WB})}$$

**Equation 15**

Where:

| | | |
|---|---|---|
| $K_{GB}$ | = | Partition coefficient of the chemical between gut and organism (unitless) |
| $K_{OW}$ | = | chemical-specific octanol-water partition coefficient (kg/L) |
| $V_{LG}$ | = | lipid fraction of the gut (unitless) |
| $V_{NG}$ | = | NLOM fraction of the gut (unitless) |
| $V_{PG}$ | = | NLOC fraction of the gut (unitless) |
| $V_{WG}$ | = | water fraction of the gut (unitless) |
| $\beta$ (BETA) | = | NLOM-octanol proportionality constant (unitless) |
| $\gamma$ (GAMMA) | = | NLOC-octanol proportionality constant (unitless) |
| $V_{LB}$ | = | lipid fraction of the organism (unitless) |
| $V_{NB}$ | = | NLOM fraction of the organism (unitless) |
| $V_{WB}$ | = | water fraction of the organism (unitless) |

These gut fractions are estimated as shown in Equation 16 through 19; collectively, they add up to a number approaching 1 and are dependent upon the assimilation efficiency fraction for each component. (Arnot and Gobas 2004). The fractions of lipid, NLOM, NLOC, and water present in the tissue of the organism ($V_{LB}$, $V_{NB}$, $V_{PG}$, and $V_{WB}$, respectively) are based on organism-specific information:

$$V_{LG} = \frac{([1-\varepsilon_L]\times V_{LD})}{(1-\varepsilon_L \times V_{LD})+(1-\varepsilon_N \times V_{ND})+([1-\varepsilon_W]\times V_{WD})}$$

**Equation 16**

$$V_{NG} = \frac{([1-\varepsilon_N]\times V_{LD})}{([1-\varepsilon_L]\times V_{LD})+([1-\varepsilon_N]\times V_{ND})+([1-\varepsilon_W]\times V_{WD})}$$

**Equation 17**

$$V_{PG} = \frac{([1-\varepsilon_P]\times V_{LD})}{([1-\varepsilon_L]\times V_{LD})+([1-\varepsilon_N]\times V_{ND})+([1-\varepsilon_P]\times V_{PD})+([1-\varepsilon_W]\times V_{WD})}$$

**Equation 18**

$$V_{WG} = \frac{([1-\varepsilon_L]\times V_{WD})}{([1-\varepsilon_L]\times V_{LD})+([1-\varepsilon_N]\times V_{ND})+([1-\varepsilon_W]\times V_{WD})}$$

**Equation 19**

Where:

| | | |
|---|---|---|
| $V_{LG}$ | = | lipid fraction of the gut (unitless) |
| $V_{NG}$ | = | NLOM fraction of the gut (unitless) |
| $V_{PG}$ | = | NLOC fraction of the gut (unitless) |
| $V_{WG}$ | = | water fraction of the gut (unitless) |
| $V_{LD}$ | = | lipid fraction of the diet (unitless) |
| $V_{ND}$ | = | NLOM fraction of the diet (unitless) |
| $V_{PD}$ | = | NLOC fraction of the diet (unitless) |
| $V_{WD}$ | = | water fraction of the diet (unitless) |
| $\varepsilon_L$ | = | dietary assimilation efficiencies of lipid (unitless) |
| $\varepsilon_N$ | = | dietary assimilation efficiencies of NLOM (unitless) |
| $\varepsilon_P$ | = | dietary assimilation efficiencies of NLOC (unitless) |
| $\varepsilon_W$ | = | dietary assimilation efficiencies of water (unitless) |

In the model, $Z_{water}$ is used to determine chemical uptake from water in the gut ($V_{WG}$), while $Z_{lipid}$ is used to determine chemical uptake from lipid matter in the gut ($V_{LG}$), non-lipid organic matter in the gut ($V_{NG}$), and non-lipid organic carbon in the gut ($V_{PG}$). These parameters are used in conjunction with the above equations to describe the chemical flux between an organism's tissue and the material in its gut (see Section 2.4.3 for the full equation).

### 2.3.4 Growth

Growth rate information is available for a wide range of species. However, growth rates may vary among and within species according to a number of factors, including, but not limited to, organism size and age, environmental temperature, and availability and quality of food (Arnot and Gobas 2004). The recommended approximation for growth rate in the absence of empirical data is (Arnot and Gobas 2004; Thomann et al. 1992) are as follows:

For temperatures around 10°C, Equation 20 is used.

$$k_G = 0.000502 \times W_B^{-0.2}$$ **Equation 20**

For temperatures around 25°C, Equation 21 is used.

$$k_G = 0.00251 \times W_B^{-0.2}$$ **Equation 21**

Where:

| | | |
|---|---|---|
| $k_G$ | = | growth rate constant (1/day) |
| $W_B$ | = | wet weight of organism (kg) |

The growth rate in the model is temperature dependent. Equation 20 is used when the temperature is less than 17.5°C, and Equation 21 is used when the temperature exceeds 17.5°C.

### 2.3.5 Metabolic biotransformation rate constant

Chemical compounds may be eliminated from an organism through metabolic biotransformation, during which the parent compound undergoes structural changes to become a chemical derivative or metabolite of the original compound. The metabolic process is species- and chemical-specific, and the inclusion of metabolic biotransformation in the mechanistic model is further discussed in Section 3.3.1 of the main report and in Appendix G.

## 2.4 Species-Specific Calculations

Many of the equations presented in Arnot and Gobas (2004) were included in the version of the model used in this LPRSA bioaccumulation modeling. Excerpts of the VBA code used to run the LPRSA mechanistic model are presented below, with definitions of each input parameter used and examples of how those parameters fit into the equations required to run the model. The parameter abbreviations used by Arnot and Gobas (2004) were altered slightly for convenience in the version presented here; however, the functionality of the model was preserved.

The entire VBA code is presented in Section 3; because of the iterative nature of the model, a representative organism from each of the two main types of organisms modeled (i.e., plankton [Section 2.4.1] and small forage fish [Section 2.4.2]) has been selected for a more detailed description. Section 3, the complete VBA code, presents the exact coding information used for the other organisms.

The identifying numbers used to represent species in the mechanistic model are presented in Table 1.

**Table 1. Identifying numbers for species used in model code**

| Identifying Number | FWM Compartment |
|---|---|
| 2 | phytoplankton/algae |
| 3 | zooplankton |
| 4 | benthic invertebrate DEPs |
| 5 | benthic invertebrate DETs |
| 6 | benthic invertebrate C/Os |
| 7 | small filter feeding fish |
| 8 | small forage fish |
| 9 | carp |
| 10 | catfish |
| 11 | white perch |
| 12 | American eel |
| 13 | bass |
| 14 | blue crab |

C/O – carnivore/omnivore
DEP – deposit feeder
DET –detritivore
FWM – food web model

This numbering methodology allowed for the identification of species-specific values within the code without having to write out the entire species name as it accompanied each of the individual parameters.

### 2.4.1 Example VBA code for phytoplankton/algae

This section presents the VBA code for the phytoplankton/algae model compartment. Definition of terms are provided in Table 2.

```
VLB2 = empirical value defined by model user
VWB2 = empirical value defined by model user
VNB2 = 0
VPB2 = empirical value defined by model user
UA = empirical value defined by model user
UB = empirical value defined by model user
K12 = 1 / (UA + (UB / KOW))
KPW2 = (VLB2 * KOW) + (VPB2 * GAMMA * KOW) + VWB2
```

```
K22 = K12 / KPW2
KG2 = Worksheets("Output").Cells(54, 5)
FPW2 = Worksheets("Output").Cells(50, 5)
CB2 = (CWB * K12 * (1 - FPW2)) / (K22 + KG2)
```

### Table 2. Definition of equation variables for phytoplankton/algae

| Acronym | Definition |
|---|---|
| VLB2 | lipid fraction of organism (unitless) |
| VNB2 | non-lipid organic matter fraction of organism (unitless) |
| VPB2 | non-lipid organic carbon fraction of organism (unitless) |
| VWB2 | water fraction of organism (unitless) |
| GAMMA | non-lipid organic carbon (NLOC) proportionality constant (unitless) |
| K12 | gill uptake rate constant ($d^{-1}$) |
| UA | uptake constant A (unitless) |
| UB | uptake constant B (unitless) |
| KOW | chemical-specific octanol-water partition coefficient (kg/L) |
| KPW2 | organism-water partition coefficient (unitless) |
| K22 | gill and skin elimination rate constant ($d^{-1}$) |
| KG2 | growth rate constant ($d^{-1}$) |
| KM2 | metabolic biotransformation rate constant ($d^{-1}$) |
| FPW2 | fraction of sediment porewater ventilated by organism (unitless) |
| CWB | biologically available concentration of chemical in water (ng/g) |
| CB2 | predicted tissue concentration in organism (ng/g) |

### 2.4.2 Example VBA code for small forage fish

This section presents the VBA code for the small forage fish model compartment. Definition of terms are provided in Table 3.

```
WB8 = empirical value defined by model user
VLB8 = empirical value defined by model user
VWB8 = empirical value defined by model user
VNB8 = empirical value defined by model user
VPB8 = empirical value defined by model user
WBL8 = WB8 * VLB8
QW8 = 88.3 * WB8 ^ 0.6
QL8 = QW8 * 0.01
    'Temperature dependent growth
If TW < 17.5 Then
    KG8 = 0.000502 * WB8 ^ -0.2
Else
    KG8 = 0.00251 * WB8 ^ -0.2
End If
GV8 = (1400 * (WB8 ^ 0.65)) / COX
GD8 = 0.022 * WB8 ^ 0.85 * Exp(0.06 * TW)
```

DF81 = dietary fraction of prey item 1 (sediment) for organism 8 (small forage fish)

DF8p = dietary fraction of prey item p (near-bottom particulates) for organism 8 (small forage fish)

DF82 = dietary fraction of prey item 2 (phytoplankton) for organism 8 (small forage fish)

DF83 = dietary fraction of prey item 3 (zooplankton) for organism 8 (small forage fish)

DF84 = dietary fraction of prey item 4 (deposit feeders) for organism 8 (small forage fish)

DF85 = dietary fraction of prey item 5 (detritivores) for organism 8 (small forage fish)

DF86 = dietary fraction of prey item 6 (benthic invertebrate – carnivore/omnivores) for organism 8 (small forage fish)

DF87 = dietary fraction of prey item 7 (filter feeding fish) for organism 8 (small forage fish)

eL8 = empirical value defined by model user

eP8 = empirical value defined by model user

eN8 = empirical value defined by model user

eW8 = empirical value defined by model user

FPW8 = empirical value defined by model user

Food8A = DF81 * VLBsed + DF8p * VLBpart + DF82 * VLB2 + DF83 * VLB3 + DF84 * VLB4 + DF85 * VLB5 + DF86 * VLB6 + DF87 * VLB7

Food8B = DF81 * VNBsed + DF8p * VNBpart + DF82 * VNB2 + DF83 * VNB3 + DF84 * VNB4 + DF85 * VNB5 + DF86 * VNB6 + DF87 * VNB7

Food8C = DF81 * VWBsed + DF8p * VWBpart + DF82 * VWB2 + DF83 * VWB3 + DF84 * VWB4 + DF85 * VWB5 + DF86 * VWB6 + DF87 * VWB7

Food8D = DF81 * VPBsed + DF8p * VPBpart + DF82 * VPB2 + DF83 * VPB3 + DF84 * VPB4 + DF85 * VPB5 + DF86 * VPB6 + DF87 * VPB7

Food8E = DF81 * CST + DF8p * CFL + DF82 * CB2 + DF83 * CB3 + DF84 * CB4 + DF85 * CB5 + DF86 * CB6 + DF87 * CB7

GF8 = (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D)) * GD8

VLG8 = ((1 - eL8) * Food8A) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D))

VNG8 = ((1 - eN8) * Food8B) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D))

VWG8 = ((1 - eW8) * Food8C) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D))

VPG8 = ((1 - eP8) * Food8D) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D))

ED8 = 1 / (EDA * KOW + EDB)

KD8 = ED8 * GD8 / WB8

EWW8 = 1 / (1.85 + (155 / KOW))

K18 = EWW8 * GV8 / WB8

KPW8 = (VLB8 * KOW) + (VNB8 * BETA * KOW) + VWB8

K28 = K18 / KPW8

Zorg8 = (VLB8 * Zlipid) + (VNB8 * BETA * Zlipid) + (VWB8 * Zwater)

Zgut8 = VLG8 * Zlipid + VNG8 * BETA * Zlipid + VPG8 * GAMMA * Zlipid + VWG8 * Zwater

KGB8 = Zgut8 / Zorg8

KE8 = KGB8 / WB8 * ED8 * GF8

CB8 = (CWB * K18 * (1 - FPW8) + CSD * K18 * FPW8 + KD8 * Food8E) / (K28 + KE8 + KG8 + KM8)

### Table 3. Definition of equation variables for small forage fish

| Acronym | Definition |
|---|---|
| VLB8 | lipid fraction of organism (unitless) |
| VNB8 | non-lipid organic matter fraction of organism (unitless) |

| | |
|---|---|
| VPB8 | non-lipid organic carbon fraction of organism (unitless) |
| VWB8 | water fraction of organism (unitless) |
| WB8 | wet weight of organism (kg) |
| WBL8 | organism lipid weight (kg) |
| QW8 | aqueous transport parameter for organism ($d^{-1}$) |
| QL8 | lipid transport parameter for organism ($d^{-1}$) |
| GD8 | food ingestion rate (kg food/day) |
| TW | mean water temperature (°C) |
| KG8 | growth rate constant ($d^{-1}$) |
| GV8 | gill ventilation rate (L/day) |
| COX | dissolved oxygen content (mg/L) |
| DF | fraction of other organism in fish diet (unitless) |
| FoodA8 | intermediate calculation term (unitless) |
| FoodB8 | intermediate calculation term (unitless) |
| FoodC8 | intermediate calculation term (unitless) |
| FoodD8 | intermediate calculation term (unitless) |
| FoodE8 | intermediate calculation term (unitless) |
| VLBsed | lipid fraction of organism (unitless); always equal to 0 |
| VNBsed | NLOM content of sediment; always equal to 0 |
| VPBsed | NLOC content of sediment (unitless) |
| VWBsed | water content of sediment (unitless) |
| GF8 | fecal egestion rate (kg food/day) |
| eL8 | lipid dietary absorption efficiency for organism (unitless) |
| eN8 | NLOM dietary absorption efficiency for organism (unitless) |
| eP8 | NLOC dietary absorption efficiency for organism (unitless) |
| eW8 | water dietary absorption efficiency for organism (unitless) |
| VLG8 | lipid fraction in organism gut (unitless) |
| VNG8 | NLOM fraction in organism gut (unitless) |
| VPG8 | NLOC fraction in organism gut (unitless) |
| VWG8 | water fraction in organism gut (unitless) |
| ED8 | intestinal tract chemical transfer efficiency (unitless) |
| KD8 | dietary uptake rate constant ($d^{-1}$) |
| EDA | dietary chemical transfer constant a |
| EDB | dietary chemical transfer constant b |
| K18 | gill uptake rate constant ($d^{-1}$) |
| EWW8 | gill chemical transfer efficiency (unitless) |
| KOW | chemical-specific octanol-water partition coefficient (kg/L) |
| KPW8 | organism-water partition coefficient (unitless) |
| K28 | gill and skin elimination rate constant ($d^{-1}$) |
| BETA | NLOM proportionality constant (unitless) |
| GAMMA | NLOC proportionality constant (unitless) |
| Zorg8 | organism fugacity capacity constant (mol m-3 Pa-1) |
| Zgut8 | organism intestinal tract fugacity capacity constant (mol m-3 Pa-1) |
| Zlipid | lipid fugacity capacity constant |
| Zwater | water fugacity capacity constant |

| | |
|---|---|
| KGB8 | gut-organism partition coefficient (unitless) |
| KE8 | fecal egestion rate constant ($d^{-1}$) |
| KM8 | metabolic biotransformation rate constant ($d^{-1}$) |
| FPW8 | fraction of sediment porewater ventilated (unitless) |
| CWB | biologically available concentration of chemical in water (ng/g) |
| CST | total concentration of chemical in sediment (ng/g) |
| CFL | total concentration of chemical in near-bottom particulates (ng/g) |
| CSD | concentration of chemical in sediment porewater (ng/g) |
| CB8 | tissue concentration in $8^{th}$ organism (ng/g) |

# 3 Complete VBA Model Code

```
Option Explicit
Option Base 1

'Define variables

Dim DT As Single
Dim KOW As Single
Dim BETA As Single
Dim GAMMA As Single
Dim EDA As Single
Dim EDB As Single
Dim TW As Single
Dim CPW As Single
Dim CWB As Single
Dim CST As Single
Dim CSD As Single
Dim CPART As Single
Dim CPART_DET As Single
Dim CFL As Single
Dim FPW2 As Single
Dim FPW3 As Single
Dim FPW4 As Single
Dim FPW5 As Single
Dim FPW6 As Single
Dim FPW7 As Single
Dim FPW8 As Single
Dim FPW9 As Single
Dim Zwater As Single
Dim Zlipid As Single
Dim VLBsed As Single
Dim VNBsed As Single
Dim VWBsed As Single
Dim VPBsed As Single
Dim VLBpart As Single
Dim VNBpart As Single
Dim VWBpart As Single
Dim VPBpart As Single
Dim VLBpart_DET As Single
Dim VNBpart_DET As Single
Dim VWBpart_DET As Single
Dim VPBpart_DET As Single
Dim COX As Single
Dim H As Single
Dim CWT As Single
Dim XPOC As Single
```

```
Dim APOC As Single
Dim XDOC As Single
Dim ADOC As Single
Dim BSF As Single
Dim chemconstant As Single
Dim OCSS As Single
Dim OCPart As Single
Dim CAC As Single
Dim CAC_DET As Single
Dim DOconstant As Single

Dim TW_mf As Single
Dim CWB_mf As Single
Dim CSD_mf As Single
Dim CST_mf As Single
Dim CPART_mf As Single
Dim CPART_DET_mf As Single
Dim CFL_mf As Single
Dim CPW_mf As Single
Dim VNBsed_mf As Single
Dim VLBsed_mf As Single
Dim VWBsed_mf As Single
Dim VPBsed_mf As Single
Dim OCPART_mf As Single
Dim VNBpart_mf As Single
Dim VLBpart_mf As Single
Dim VWBpart_mf As Single
Dim VPBpart_mf As Single
Dim VNBpart_DET_mf As Single
Dim VLBpart_DET_mf As Single
Dim VWBpart_DET_mf As Single
Dim VPBpart_DET_mf As Single
Dim COX_mf As Single

'----
Dim WB2 As Single
Dim VLB2 As Single
Dim VNB2 As Single
Dim VWB2 As Single
Dim VPB2 As Single
Dim UA As Single
Dim UB As Single
Dim K12 As Single
Dim K22 As Single
Dim KPW2 As Single
Dim KG2 As Single
Dim CB2 As Single
'----
```

```
Dim WB3 As Single
Dim VLB3 As Single
Dim VNB3 As Single
Dim VWB3 As Single
Dim VPB3 As Single
Dim WBL3 As Single
Dim KM3 As Single
Dim QW3 As Single
Dim QL3 As Single
Dim GD3 As Single
Dim KG3 As Single
Dim GV3 As Single
Dim DF32 As Single
Dim eL3 As Single
Dim eN3 As Single
Dim eW3 As Single
Dim eP3 As Single
Dim GF3 As Single
Dim VLG3 As Single
Dim VNG3 As Single
Dim VWG3 As Single
Dim VPG3 As Single
Dim ED3 As Single
Dim KD3 As Single
Dim EWW3 As Single
Dim K13 As Single
Dim KPW3 As Single
Dim K23 As Single
Dim KE3 As Single
Dim Food3A As Single
Dim Food3B As Single
Dim Food3C As Single
Dim Food3D As Single
Dim Zorg3 As Single
Dim Zgut3 As Single
Dim KGB3 As Single
Dim CB3 As Single
'----
Dim WB4 As Single
Dim VLB4 As Single
Dim VNB4 As Single
Dim VWB4 As Single
Dim VPB4 As Single
Dim WBL4 As Single
Dim KM4 As Single
Dim QW4 As Single
Dim QL4 As Single
Dim GD4 As Single
```

```
Dim KG4 As Single
Dim GV4 As Single
Dim DF41 As Single
Dim DF4p As Single
Dim DF42 As Single
Dim DF43 As Single
Dim eL4 As Single
Dim eN4 As Single
Dim eW4 As Single
Dim eP4 As Single
Dim GF4 As Single
Dim VLG4 As Single
Dim VNG4 As Single
Dim VWG4 As Single
Dim VPG4 As Single
Dim ED4 As Single
Dim KD4 As Single
Dim EWW4 As Single
Dim K14 As Single
Dim KPW4 As Single
Dim K24 As Single
Dim KE4 As Single
Dim SCV4 As Single
Dim Food4A As Single
Dim Food4B As Single
Dim Food4C As Single
Dim Food4D As Single
Dim Food4E As Single
Dim Zorg4 As Single
Dim Zgut4 As Single
Dim KGB4 As Single
Dim CB4 As Single
'----
Dim WB5 As Single
Dim VLB5 As Single
Dim VNB5 As Single
Dim VWB5 As Single
Dim VPB5 As Single
Dim WBL5 As Single
Dim KM5 As Single
Dim QW5 As Single
Dim QL5 As Single
Dim GD5 As Single
Dim KG5 As Single
Dim GV5 As Single
Dim DF51 As Single
Dim DF5p As Single
Dim DF52 As Single
```

```
Dim DF53 As Single
Dim DF54 As Single
Dim eL5 As Single
Dim eN5 As Single
Dim eW5 As Single
Dim eP5 As Single
Dim GF5 As Single
Dim VLG5 As Single
Dim VNG5 As Single
Dim VWG5 As Single
Dim VPG5 As Single
Dim ED5 As Single
Dim KD5 As Single
Dim EWW5 As Single
Dim K15 As Single
Dim KPW5 As Single
Dim K25 As Single
Dim KE5 As Single
Dim Food5A As Single
Dim Food5B As Single
Dim Food5C As Single
Dim Food5D As Single
Dim Food5E As Single
Dim Zorg5 As Single
Dim Zgut5 As Single
Dim KGB5 As Single
Dim CB5 As Single
'----
Dim WB6 As Single
Dim VLB6 As Single
Dim VNB6 As Single
Dim VWB6 As Single
Dim VPB6 As Single
Dim WBL6 As Single
Dim KM6 As Single
Dim QW6 As Single
Dim QL6 As Single
Dim GD6 As Single
Dim KG6 As Single
Dim GV6 As Single
Dim DF61 As Single
Dim DF6p As Single
Dim DF62 As Single
Dim DF63 As Single
Dim DF64 As Single
Dim DF65 As Single
Dim eL6 As Single
Dim eN6 As Single
```

```
Dim eW6 As Single
Dim eP6 As Single
Dim GF6 As Single
Dim VLG6 As Single
Dim VNG6 As Single
Dim VWG6 As Single
Dim VPG6 As Single
Dim ED6 As Single
Dim KD6 As Single
Dim EWW6 As Single
Dim K16 As Single
Dim KPW6 As Single
Dim K26 As Single
Dim KE6 As Single
Dim Food6A As Single
Dim Food6B As Single
Dim Food6C As Single
Dim Food6D As Single
Dim Food6E As Single
Dim Zorg6 As Single
Dim Zgut6 As Single
Dim KGB6 As Single
Dim CB6 As Single
'----
Dim WB7 As Single
Dim VLB7 As Single
Dim VNB7 As Single
Dim VWB7 As Single
Dim VPB7 As Single
Dim WBL7 As Single
Dim KM7 As Single
Dim QW7 As Single
Dim QL7 As Single
Dim GD7 As Single
Dim KG7 As Single
Dim GV7 As Single
Dim DF71 As Single
Dim DF7p As Single
Dim DF72 As Single
Dim DF73 As Single
Dim DF74 As Single
Dim DF75 As Single
Dim DF76 As Single
Dim eL7 As Single
Dim eN7 As Single
Dim eW7 As Single
Dim eP7 As Single
Dim GF7 As Single
```

```
Dim VLG7 As Single
Dim VNG7 As Single
Dim VWG7 As Single
Dim VPG7 As Single
Dim ED7 As Single
Dim KD7 As Single
Dim EWW7 As Single
Dim K17 As Single
Dim KPW7 As Single
Dim K27 As Single
Dim KE7 As Single
Dim Food7A As Single
Dim Food7B As Single
Dim Food7C As Single
Dim Food7D As Single
Dim Food7E As Single
Dim Zorg7 As Single
Dim Zgut7 As Single
Dim KGB7 As Single
Dim CB7 As Single
'----
Dim WB8 As Single
Dim VLB8 As Single
Dim VNB8 As Single
Dim VWB8 As Single
Dim VPB8 As Single
Dim WBL8 As Single
Dim KM8 As Single
Dim QW8 As Single
Dim QL8 As Single
Dim GD8 As Single
Dim KG8 As Single
Dim GV8 As Single
Dim DF81 As Single
Dim DF8p As Single
Dim DF82 As Single
Dim DF83 As Single
Dim DF84 As Single
Dim DF85 As Single
Dim DF86 As Single
Dim DF87 As Single
Dim eL8 As Single
Dim eN8 As Single
Dim eW8 As Single
Dim eP8 As Single
Dim GF8 As Single
Dim VLG8 As Single
Dim VNG8 As Single
```

```
Dim VWG8 As Single
Dim VPG8 As Single
Dim ED8 As Single
Dim KD8 As Single
Dim EWW8 As Single
Dim K18 As Single
Dim KPW8 As Single
Dim K28 As Single
Dim KE8 As Single
Dim Food8A As Single
Dim Food8B As Single
Dim Food8C As Single
Dim Food8D As Single
Dim Food8E As Single
Dim Zorg8 As Single
Dim Zgut8 As Single
Dim KGB8 As Single
Dim CB8 As Single
'----
Dim WB9 As Single
Dim VLB9 As Single
Dim VNB9 As Single
Dim VWB9 As Single
Dim VPB9 As Single
Dim WBL9 As Single
Dim KM9 As Single
Dim QW9 As Single
Dim QL9 As Single
Dim GD9 As Single
Dim KG9 As Single
Dim GV9 As Single
Dim DF91 As Single
Dim DF9p As Single
Dim DF92 As Single
Dim DF93 As Single
Dim DF94 As Single
Dim DF95 As Single
Dim DF96 As Single
Dim DF97 As Single
Dim DF98 As Single
Dim eL9 As Single
Dim eN9 As Single
Dim eW9 As Single
Dim eP9 As Single
Dim GF9 As Single
Dim VLG9 As Single
Dim VNG9 As Single
Dim VWG9 As Single
```

```
Dim VPG9 As Single
Dim ED9 As Single
Dim KD9 As Single
Dim EWW9 As Single
Dim K19 As Single
Dim KPW9 As Single
Dim K29 As Single
Dim KE9 As Single
Dim Food9A As Single
Dim Food9B As Single
Dim Food9C As Single
Dim Food9D As Single
Dim Food9E As Single
Dim Zorg9 As Single
Dim Zgut9 As Single
Dim KGB9 As Single
Dim CB9 As Single
'----
Dim WB10 As Single
Dim VLB10 As Single
Dim VNB10 As Single
Dim VWB10 As Single
Dim VPB10 As Single
Dim WBL10 As Single
Dim KM10 As Single
Dim QW10 As Single
Dim QL10 As Single
Dim GD10 As Single
Dim KG10 As Single
Dim GV10 As Single
Dim DF101 As Single
Dim DF10p As Single
Dim DF102 As Single
Dim DF103 As Single
Dim DF104 As Single
Dim DF105 As Single
Dim DF106 As Single
Dim DF107 As Single
Dim DF108 As Single
Dim DF109 As Single
Dim eL10 As Single
Dim eN10 As Single
Dim eW10 As Single
Dim eP10 As Single
Dim GF10 As Single
Dim VLG10 As Single
Dim VNG10 As Single
Dim VWG10 As Single
```

```
Dim VPG10 As Single
Dim ED10 As Single
Dim KD10 As Single
Dim EWW10 As Single
Dim K110 As Single
Dim KPW10 As Single
Dim K210 As Single
Dim KE10 As Single
Dim FPW10 As Single
Dim Food10A As Single
Dim Food10B As Single
Dim Food10C As Single
Dim Food10D As Single
Dim Food10E As Single
Dim Zorg10 As Single
Dim Zgut10 As Single
Dim KGB10 As Single
Dim CB10 As Single
'----
Dim WB11 As Single
Dim VLB11 As Single
Dim VNB11 As Single
Dim VWB11 As Single
Dim VPB11 As Single
Dim WBL11 As Single
Dim KM11 As Single
Dim QW11 As Single
Dim QL11 As Single
Dim GD11 As Single
Dim KG11 As Single
Dim GV11 As Single
Dim DF111 As Single
Dim DF11p As Single
Dim DF112 As Single
Dim DF113 As Single
Dim DF114 As Single
Dim DF115 As Single
Dim DF116 As Single
Dim DF117 As Single
Dim DF118 As Single
Dim DF119 As Single
Dim DF1110 As Single
Dim eL11 As Single
Dim eN11 As Single
Dim eW11 As Single
Dim eP11 As Single
Dim GF11 As Single
Dim VLG11 As Single
```

```
Dim VNG11 As Single
Dim VWG11 As Single
Dim VPG11 As Single
Dim ED11 As Single
Dim KD11 As Single
Dim EWW11 As Single
Dim K111 As Single
Dim KPW11 As Single
Dim K211 As Single
Dim KE11 As Single
Dim FPW11 As Single
Dim Food11A As Single
Dim Food11B As Single
Dim Food11C As Single
Dim Food11D As Single
Dim Food11E As Single
Dim Zorg11 As Single
Dim Zgut11 As Single
Dim KGB11 As Single
Dim CB11 As Single
'----
Dim WB12 As Single
Dim VLB12 As Single
Dim VNB12 As Single
Dim VWB12 As Single
Dim VPB12 As Single
Dim WBL12 As Single
Dim KM12 As Single
Dim QW12 As Single
Dim QL12 As Single
Dim GD12 As Single
Dim KG12 As Single
Dim GV12 As Single
Dim DF121 As Single
Dim DF12p As Single
Dim DF122 As Single
Dim DF123 As Single
Dim DF124 As Single
Dim DF125 As Single
Dim DF126 As Single
Dim DF127 As Single
Dim DF128 As Single
Dim DF129 As Single
Dim DF1210 As Single
Dim DF1211 As Single
Dim eL12 As Single
Dim eN12 As Single
Dim eW12 As Single
```

```
Dim eP12 As Single
Dim GF12 As Single
Dim VLG12 As Single
Dim VNG12 As Single
Dim VWG12 As Single
Dim VPG12 As Single
Dim ED12 As Single
Dim KD12 As Single
Dim EWW12 As Single
Dim K112 As Single
Dim KPW12 As Single
Dim K212 As Single
Dim KE12 As Single
Dim FPW12 As Single
Dim Food12A As Single
Dim Food12B As Single
Dim Food12C As Single
Dim Food12D As Single
Dim Food12E As Single
Dim Zorg12 As Single
Dim Zgut12 As Single
Dim KGB12 As Single
Dim CB12 As Single
'----
Dim WB13 As Single
Dim VLB13 As Single
Dim VNB13 As Single
Dim VWB13 As Single
Dim VPB13 As Single
Dim WBL13 As Single
Dim KM13 As Single
Dim QW13 As Single
Dim QL13 As Single
Dim GD13 As Single
Dim KG13 As Single
Dim GV13 As Single
Dim DF131 As Single
Dim DF13p As Single
Dim DF132 As Single
Dim DF133 As Single
Dim DF134 As Single
Dim DF135 As Single
Dim DF136 As Single
Dim DF137 As Single
Dim DF138 As Single
Dim DF139 As Single
Dim DF1310 As Single
Dim DF1311 As Single
```

```
Dim DF1312 As Single
Dim eL13 As Single
Dim eN13 As Single
Dim eW13 As Single
Dim eP13 As Single
Dim GF13 As Single
Dim VLG13 As Single
Dim VNG13 As Single
Dim VWG13 As Single
Dim VPG13 As Single
Dim ED13 As Single
Dim KD13 As Single
Dim EWW13 As Single
Dim K113 As Single
Dim KPW13 As Single
Dim K213 As Single
Dim KE13 As Single
Dim FPW13 As Single
Dim Food13A As Single
Dim Food13B As Single
Dim Food13C As Single
Dim Food13D As Single
Dim Food13E As Single
Dim Zorg13 As Single
Dim Zgut13 As Single
Dim KGB13 As Single
Dim CB13 As Single
'----
Dim WB14 As Single
Dim VLB14 As Single
Dim VNB14 As Single
Dim VWB14 As Single
Dim VPB14 As Single
Dim WBL14 As Single
Dim KM14 As Single
Dim QW14 As Single
Dim QL14 As Single
Dim GD14 As Single
Dim KG14 As Single
Dim GV14 As Single
Dim DF141 As Single
Dim DF14p As Single
Dim DF142 As Single
Dim DF143 As Single
Dim DF144 As Single
Dim DF145 As Single
Dim DF146 As Single
Dim DF147 As Single
```

```
Dim DF148 As Single
Dim DF149 As Single
Dim DF1410 As Single
Dim DF1411 As Single
Dim DF1412 As Single
Dim DF1413 As Single
Dim eL14 As Single
Dim eN14 As Single
Dim eW14 As Single
Dim eP14 As Single
Dim GF14 As Single
Dim VLG14 As Single
Dim VNG14 As Single
Dim VWG14 As Single
Dim VPG14 As Single
Dim ED14 As Single
Dim KD14 As Single
Dim EWW14 As Single
Dim K114 As Single
Dim KPW14 As Single
Dim K214 As Single
Dim KE14 As Single
Dim FPW14 As Single
Dim Food14A As Single
Dim Food14B As Single
Dim Food14C As Single
Dim Food14D As Single
Dim Food14E As Single
Dim Zorg14 As Single
Dim Zgut14 As Single
Dim KGB14 As Single
Dim CB14 As Single

'Mudflat variables

Dim CB2_mf As Single
Dim GD3_mf As Single
Dim GF3_mf As Single
Dim KD3_mf As Single
Dim KE3_mf As Single
Dim CB3_mf As Single
Dim GD4_mf As Single
Dim Food4B_mf As Single
Dim Food4C_mf As Single
Dim Food4D_mf As Single
Dim Food4E_mf As Single
Dim GF4_mf As Single
Dim VLG4_mf As Single
```

```
Dim VPG4_mf As Single
Dim VNG4_mf As Single
Dim VWG4_mf As Single
Dim KD4_mf As Single
Dim Zgut4_mf As Single
Dim KGB4_mf As Single
Dim KE4_mf As Single
Dim CB4_mf As Single
Dim GD5_mf As Single
Dim Food5B_mf As Single
Dim Food5C_mf As Single
Dim Food5D_mf As Single
Dim Food5E_mf As Single
Dim GF5_mf As Single
Dim VLG5_mf As Single
Dim VPG5_mf As Single
Dim VNG5_mf As Single
Dim VWG5_mf As Single
Dim KD5_mf As Single
Dim Zgut5_mf As Single
Dim KGB5_mf As Single
Dim KE5_mf As Single
Dim CB5_mf As Single
Dim GD6_mf As Single
Dim Food6B_mf As Single
Dim Food6C_mf As Single
Dim Food6D_mf As Single
Dim Food6E_mf As Single
Dim GF6_mf As Single
Dim VLG6_mf As Single
Dim VPG6_mf As Single
Dim VNG6_mf As Single
Dim VWG6_mf As Single
Dim KD6_mf As Single
Dim Zgut6_mf As Single
Dim KGB6_mf As Single
Dim KE6_mf As Single
Dim CB6_mf As Single
Dim GD7_mf As Single
Dim Food7B_mf As Single
Dim Food7C_mf As Single
Dim Food7D_mf As Single
Dim Food7E_mf As Single
Dim GF7_mf As Single
Dim VLG7_mf As Single
Dim VPG7_mf As Single
Dim VNG7_mf As Single
Dim VWG7_mf As Single
```

```
Dim KD7_mf As Single
Dim Zgut7_mf As Single
Dim KGB7_mf As Single
Dim KE7_mf As Single
Dim CB7_mf As Single
Dim GD8_mf As Single
Dim Food8B_mf As Single
Dim Food8C_mf As Single
Dim Food8D_mf As Single
Dim Food8E_mf As Single
Dim GF8_mf As Single
Dim VLG8_mf As Single
Dim VPG8_mf As Single
Dim VNG8_mf As Single
Dim VWG8_mf As Single
Dim KD8_mf As Single
Dim Zgut8_mf As Single
Dim KGB8_mf As Single
Dim KE8_mf As Single
Dim CB8_mf As Single
Dim GD9_mf As Single
Dim Food9B_mf As Single
Dim Food9C_mf As Single
Dim Food9D_mf As Single
Dim Food9E_mf As Single
Dim GF9_mf As Single
Dim VLG9_mf As Single
Dim VPG9_mf As Single
Dim VNG9_mf As Single
Dim VWG9_mf As Single
Dim KD9_mf As Single
Dim Zgut9_mf As Single
Dim KGB9_mf As Single
Dim KE9_mf As Single
Dim CB9_mf As Single
Dim GV3_mf As Single
Dim GV4_mf As Single
Dim GV5_mf As Single
Dim GV6_mf As Single
Dim GV7_mf As Single
Dim GV8_mf As Single
Dim GV9_mf As Single
Dim K13_mf As Single
Dim K14_mf As Single
Dim K15_mf As Single
Dim K16_mf As Single
Dim K17_mf As Single
Dim K18_mf As Single
```

```
Dim K19_mf As Single
Dim K23_mf As Single
Dim K24_mf As Single
Dim K25_mf As Single
Dim K26_mf As Single
Dim K27_mf As Single
Dim K28_mf As Single
Dim K29_mf As Single

Dim MFon As Single
Dim SFFC As Single
Dim CPV As Single


'Added to store values between sub and function
Private dic As Scripting.Dictionary

'Value key constants
Private Const PHYTOPLANKTON As String = "phytoplankton"
Private Const ZOOPLANKTON As String = "zooplankton"
Private Const COMPARTMENT4 As String = "compartment4"
Private Const COMPARTMENT5 As String = "compartment5"
Private Const COMPARTMENT6 As String = "compartment6"
Private Const COMPARTMENT7 As String = "compartment7"
Private Const COMPARTMENT8 As String = "compartment8"
Private Const COMPARTMENT9 As String = "compartment9"
Private Const COMPARTMENT10 As String = "compartment10"
Private Const COMPARTMENT11 As String = "compartment11"
Private Const COMPARTMENT12 As String = "compartment12"
Private Const COMPARTMENT13 As String = "compartment13"
Private Const COMPARTMENT14 As String = "compartment14"

Private Const PHYTOPLANKTON_mf As String = "phytoplankton_mf"
Private Const ZOOPLANKTON_mf As String = "zooplankton_mf"
Private Const COMPARTMENT4_mf As String = "compartment4_mf"
Private Const COMPARTMENT5_mf As String = "compartment5_mf"
Private Const COMPARTMENT6_mf As String = "compartment6_mf"
Private Const COMPARTMENT7_mf As String = "compartment7_mf"
Private Const COMPARTMENT8_mf As String = "compartment8_mf"
Private Const COMPARTMENT9_mf As String = "compartment9_mf"


'Function to calculate tissue concentration for specific species and chemical

Public Function TissueConcentration(species As String, chemconstant As Integer) As Single

'Chemical Specific Parameters
  KOW = 10 ^ Worksheets("Output").Cells(4, 4 + chemconstant)
```

```
    CST = Worksheets("Output").Cells(6, 4 + chemconstant)
    CPART = Worksheets("Output").Cells(7, 4 + chemconstant)
    CWB = Worksheets("Output").Cells(8, 4 + chemconstant)
    CSD = Worksheets("Output").Cells(9, 4 + chemconstant)
    CFL = Worksheets("Output").Cells(10, 4 + chemconstant)

    CST_mf = Worksheets("Output").Cells(6, 17 + chemconstant)
    CPART_mf = Worksheets("Output").Cells(7, 17 + chemconstant)
    CWB_mf = Worksheets("Output").Cells(8, 17 + chemconstant)
    CSD_mf = Worksheets("Output").Cells(9, 17 + chemconstant)
    CFL_mf = Worksheets("Output").Cells(10, 17 + chemconstant)
    CPV = Worksheets("Output").Cells(51 + chemconstant / 4, 12)
    KM3 = Worksheets("Output").Cells(55 + chemconstant, 6)
    KM4 = Worksheets("Output").Cells(55 + chemconstant, 7)
    KM5 = Worksheets("Output").Cells(55 + chemconstant, 8)
    KM6 = Worksheets("Output").Cells(55 + chemconstant, 9)
    KM7 = Worksheets("Output").Cells(55 + chemconstant, 10)
    KM8 = Worksheets("Output").Cells(55 + chemconstant, 11)
    KM9 = Worksheets("Output").Cells(55 + chemconstant, 12)
    KM10 = Worksheets("Output").Cells(55 + chemconstant, 13)
    KM11 = Worksheets("Output").Cells(55 + chemconstant, 14)
    KM12 = Worksheets("Output").Cells(55 + chemconstant, 15)
    KM13 = Worksheets("Output").Cells(55 + chemconstant, 16)
    KM14 = Worksheets("BC").Cells(19 + chemconstant, 5)

'Call tissue concentration calculation Subroutine (runs code below)
    Call PHFWPRG5

'Write the applicable value to the cell
    TissueConcentration = CSng(dic.Item(species))

End Function

Sub PHFWPRG5()

    Set dic = New Scripting.Dictionary

'INPUT GENERAL PARAMETERS
    VLBsed = 0
    VNBsed = 0
    VPBsed = Worksheets("Output").Cells(15, 4)
    VWBsed = 1 - VPBsed

    VLBsed_mf = 0
    VNBsed_mf = 0
    VPBsed_mf = Worksheets("Output").Cells(15, 5)
    VWBsed_mf = 1 - VPBsed_mf
```

```
VLBpart = 0
VNBpart = 0
VPBpart = Worksheets("Output").Cells(16, 4)
VWBpart = 1 - VPBpart

VLBpart_mf = 0
VNBpart_mf = 0
VPBpart_mf = Worksheets("Output").Cells(16, 5)
VWBpart_mf = 1 - VPBpart_mf

VLBpart_DET = 0
VNBpart_DET = 0
VPBpart_DET = Worksheets("Output").Cells(17, 4)
VWBpart_DET = 1 - VPBpart_DET

VLBpart_DET_mf = 0
VNBpart_DET_mf = 0
VPBpart_DET_mf = Worksheets("Output").Cells(17, 5)
VWBpart_DET_mf = 1 - VPBpart_DET_mf

TW = Worksheets("Output").Cells(18, 4)
TW_mf = Worksheets("Output").Cells(18, 5)
DOconstant = Worksheets("Index").Cells(3, 1)
COX = (-0.24 * TW + 14.04) * Worksheets("Index").Cells(4 + DOconstant, 3)
COX_mf = (-0.24 * TW_mf + 14.04) * Worksheets("Index").Cells(4 + DOconstant, 3)
EDA = Worksheets("Output").Cells(35, 4)
EDB = Worksheets("Output").Cells(36, 4)
BETA = Worksheets("Output").Cells(37, 4)
GAMMA = Worksheets("Output").Cells(38, 4)

'Henry's Law constant - for the purposes of the bioaccumulation model, this parameter cancels out, and thus
    an arbitrary value of 1 is used in the model.
'The correct value at 25°C for 2378-TCDD is 1.64 Pa m3/mol and for tetraCB is 2.29 Pa m3/mol.[1]
H = 1

Zwater = 1 / H
Zlipid = Zwater * KOW


'-------------------------------
'Mudflat Calculations
'-------------------------------

'-----------------
'PHYTOPLANKTON (2)
```

[1] The sources for the Henry's law constants provided are EPA's technical factsheet on dioxin (EPA 2014) and is Shiu and Mackay (1986) for tetraCB. Both values are based on a temperature of 25°C.

```
'-----------------

    VLB2 = Worksheets("Output").Cells(42, 5)
    VWB2 = Worksheets("Output").Cells(43, 5)
    VNB2 = 0
    VPB2 = Worksheets("Output").Cells(44, 5)
    UA = Worksheets("Output").Cells(33, 4)
    UB = Worksheets("Output").Cells(34, 4)
    K12 = 1 / (UA + (UB / KOW))
    KPW2 = (VLB2 * KOW) + (VPB2 * GAMMA * KOW) + VWB2
    K22 = K12 / KPW2
    KG2 = Worksheets("Output").Cells(54, 5)
    FPW2 = Worksheets("Output").Cells(50, 5)
    CB2_mf = (CWB_mf * K12 * (1 - FPW2)) / (K22 + KG2)

    'add check value to Index worksheet
    dic.Add PHYTOPLANKTON_mf, CB2_mf

'---------------
'ZOOPLANKTON (3)
'---------------

    WB3 = Worksheets("Output").Cells(41, 6)
    VLB3 = Worksheets("Output").Cells(42, 6)
    VWB3 = Worksheets("Output").Cells(43, 6)
    VNB3 = Worksheets("Output").Cells(45, 6)
    VPB3 = 0
    WBL3 = WB3 * VLB3
    QW3 = 88.3 * WB3 ^ 0.6
    QL3 = QW3 * 0.01
    GD3_mf = 0.022 * WB3 ^ 0.85 * Exp(0.06 * TW_mf)

    'Temperature dependent growth
    If TW_mf < 17.5 Then
        KG3 = 0.000502 * WB3 ^ -0.2
    Else
        KG3 = 0.00251 * WB3 ^ -0.2
    End If

    GV3_mf = (1400 * (WB3 ^ 0.65)) / COX_mf
    DF32 = Worksheets("Diet").Cells(4, 4)
    eL3 = Worksheets("Output").Cells(46, 6)
    eP3 = Worksheets("Output").Cells(47, 6)
    eN3 = Worksheets("Output").Cells(48, 6)
    eW3 = Worksheets("Output").Cells(49, 6)
    Food3A = DF32 * VLB2
    Food3B = DF32 * VNB2
    Food3C = DF32 * VWB2
```

```
    Food3D = DF32 * VPB2
    GF3_mf = (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3) * Food3D)) *
        GD3_mf
    VLG3 = ((1 - eL3) * Food3A) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    VNG3 = ((1 - eN3) * Food3B) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    VWG3 = ((1 - eW3) * Food3C) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 -
        eP3) * Food3D))
    VPG3 = ((1 - eP3) * Food3D) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    ED3 = 1 / (EDA * KOW + EDB)
    KD3_mf = ED3 * GD3_mf / WB3
    EWW3 = 1 / (1.85 + (155 / KOW))
    K13_mf = EWW3 * GV3_mf / WB3
    KPW3 = (VLB3 * KOW) + (VNB3 * BETA * KOW) + VWB3
    K23_mf = K13_mf / KPW3
    Zorg3 = (VLB3 * Zlipid) + (VNB3 * BETA * Zlipid) + (VWB3 * Zwater)
    Zgut3 = VLG3 * Zlipid + VNG3 * BETA * Zlipid + VPG3 * GAMMA * Zlipid + VWG3 * Zwater
    KGB3 = Zgut3 / Zorg3
    KE3_mf = KGB3 / WB3 * ED3 * GF3_mf
    FPW3 = Worksheets("Output").Cells(50, 6)
    CB3_mf = (CWB_mf * K13_mf * (1 - FPW3) + CB2_mf * KD3_mf * DF32) / (K23_mf + KE3_mf + KG3 + KM3)

    'add check value to Index worksheet
    dic.Add ZOOPLANKTON_mf, CB3_mf

'-----------------------------------------
'COMPARTMENT4 (DEP - benthic deposit feeder)
'-----------------------------------------

    WB4 = Worksheets("Output").Cells(41, 7)
    VLB4 = Worksheets("Output").Cells(42, 7)
    VWB4 = Worksheets("Output").Cells(43, 7)
    VNB4 = Worksheets("Output").Cells(45, 7)
    VPB4 = 0
    WBL4 = WB4 * VLB4
    QW4 = 88.3 * WB4 ^ 0.6
    QL4 = QW4 * 0.01

    'Temperature dependent growth
    If TW_mf < 17.5 Then
      KG4 = 0.000502 * WB4 ^ -0.2
    Else
      KG4 = 0.00251 * WB4 ^ -0.2
    End If

    GV4_mf = (1400 * (WB4 ^ 0.65)) / COX_mf
    GD4_mf = 0.022 * WB4 ^ 0.85 * Exp(0.06 * TW_mf)
```

```
    DF41 = Worksheets("Diet").Cells(5, 2)
    DF4p = Worksheets("Diet").Cells(5, 3)
    DF42 = Worksheets("Diet").Cells(5, 4)
    DF43 = Worksheets("Diet").Cells(5, 5)
    eL4 = Worksheets("Output").Cells(46, 7)
    eP4 = Worksheets("Output").Cells(47, 7)
    eN4 = Worksheets("Output").Cells(48, 7)
    eW4 = Worksheets("Output").Cells(49, 7)
    FPW4 = Worksheets("Output").Cells(50, 7)
    Food4A = DF41 * VLBsed_mf + DF4p * VLBpart_DET_mf + DF42 * VLB2 + DF43 * VLB3
    Food4B_mf = DF41 * VNBsed_mf + DF4p * VNBpart_DET_mf + DF42 * VNB2 + DF43 * VNB3
    Food4C_mf = DF41 * VWBsed_mf + DF4p * VWBpart_DET_mf + DF42 * VWB2 + DF43 * VWB3
    Food4D_mf = DF41 * VPBsed_mf + DF4p * VPBpart_DET_mf + DF42 * VPB2 + DF43 * VPB3
    Food4E_mf = DF41 * CST_mf + DF4p * CFL_mf + DF42 * CB2_mf + DF43 * CB3_mf
    GF4_mf = (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B_mf) + ((1 - eW4) * Food4C_mf) + ((1 - eP4) *
        Food4D_mf)) * GD4_mf
    VLG4_mf = ((1 - eL4) * Food4A) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B_mf) + ((1 - eW4) * Food4C_mf)
        + ((1 - eP4) * Food4D_mf))
    VNG4_mf = ((1 - eN4) * Food4B_mf) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B_mf) + ((1 - eW4) *
        Food4C_mf) + ((1 - eP4) * Food4D_mf))
    VWG4_mf = ((1 - eW4) * Food4C_mf) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B_mf) + ((1 - eW4) *
        Food4C_mf) + ((1 - eP4) * Food4D_mf))
    VPG4_mf = ((1 - eP4) * Food4D_mf) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B_mf) + ((1 - eW4) *
        Food4C_mf) + ((1 - eP4) * Food4D_mf))
    ED4 = 1 / (EDA * KOW + EDB)
    KD4_mf = ED4 * GD4_mf / WB4
    EWW4 = 1 / (1.85 + (155 / KOW))
    K14_mf = EWW4 * GV4_mf / WB4
    KPW4 = (VLB4 * KOW) + (VNB4 * BETA * KOW) + VWB4
    K24_mf = K14_mf / KPW4
    Zorg4 = (VLB4 * Zlipid) + (VNB4 * BETA * Zlipid) + (VWB4 * Zwater)
    Zgut4_mf = VLG4_mf * Zlipid + VNG4_mf * BETA * Zlipid + VPG4_mf * GAMMA * Zlipid + VWG4_mf * Zwater
    KGB4_mf = Zgut4_mf / Zorg4
    KE4_mf = KGB4_mf / WB4 * ED4 * GF4_mf
    CB4_mf = (CWB_mf * K14_mf * (1 - FPW4) + K14_mf * FPW4 * CSD_mf + KD4_mf * Food4E_mf) / (K24_mf +
        KE4_mf + KG4 + KM4)

    'add check value to Index worksheet
    dic.Add COMPARTMENT4_mf, CB4_mf

'-------------------------------------------
'COMPARTMENT5 (DET - benthic detritivore)
'-------------------------------------------

    WB5 = Worksheets("Output").Cells(41, 8)
    VLB5 = Worksheets("Output").Cells(42, 8)
    VWB5 = Worksheets("Output").Cells(43, 8)
    VNB5 = Worksheets("Output").Cells(45, 8)
    VPB5 = 0
```

```
WBL5 = WB5 * VLB5
QW5 = 88.3 * WB5 ^ 0.6
QL5 = QW5 * 0.01

'Temperature dependent growth
If TW_mf < 17.5 Then
   KG5 = 0.000502 * WB5 ^ -0.2
Else
   KG5 = 0.00251 * WB5 ^ -0.2
End If

GV5_mf = (1400 * (WB5 ^ 0.65)) / COX_mf
GD5_mf = 0.022 * WB5 ^ 0.85 * Exp(0.06 * TW_mf)
DF51 = Worksheets("Diet").Cells(6, 2)
DF5p = Worksheets("Diet").Cells(6, 3)
DF52 = Worksheets("Diet").Cells(6, 4)
DF53 = Worksheets("Diet").Cells(6, 5)
DF54 = Worksheets("Diet").Cells(6, 6)
eL5 = Worksheets("Output").Cells(46, 8)
eP5 = Worksheets("Output").Cells(47, 8)
eN5 = Worksheets("Output").Cells(48, 8)
eW5 = Worksheets("Output").Cells(49, 8)
FPW5 = Worksheets("Output").Cells(50, 8)
Food5A = DF51 * VLBsed + DF5p * VLBpart_DET_mf + DF52 * VLB2 + DF53 * VLB3 + DF54 * VLB4
Food5B_mf = DF51 * VNBsed_mf + DF5p * VNBpart_DET_mf + DF52 * VNB2 + DF53 * VNB3 + DF54 * VNB4
Food5C_mf = DF51 * VWBsed_mf + DF5p * VWBpart_DET_mf + DF52 * VWB2 + DF53 * VWB3 + DF54 *
    VWB4
Food5D_mf = DF51 * VPBsed_mf + DF5p * VPBpart_DET_mf + DF52 * VPB2 + DF53 * VPB3 + DF54 * VPB4
Food5E_mf = DF51 * CST_mf + DF5p * CFL_mf + DF52 * CB2_mf + DF53 * CB3_mf + DF54 * CB4_mf
GF5_mf = (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B_mf) + ((1 - eW5) * Food5C_mf) + ((1 - eP5) *
    Food5D_mf)) * GD5_mf
VLG5_mf = ((1 - eL5) * Food5A) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B_mf) + ((1 - eW5) * Food5C_mf)
    + ((1 - eP5) * Food5D_mf))
VNG5_mf = ((1 - eN5) * Food5B_mf) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B_mf) + ((1 - eW5) *
    Food5C_mf) + ((1 - eP5) * Food5D_mf))
VWG5_mf = ((1 - eW5) * Food5C_mf) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B_mf) + ((1 - eW5) *
    Food5C_mf) + ((1 - eP5) * Food5D_mf))
VPG5_mf = ((1 - eP5) * Food5D_mf) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B_mf) + ((1 - eW5) *
    Food5C_mf) + ((1 - eP5) * Food5D_mf))
ED5 = 1 / (EDA * KOW + EDB)
KD5_mf = ED5 * GD5_mf / WB5
EWW5 = 1 / (1.85 + (155 / KOW))
K15_mf = EWW5 * GV5_mf / WB5
KPW5 = (VLB5 * KOW) + (VNB5 * BETA * KOW) + VWB5
K25_mf = K15_mf / KPW5
Zorg5 = (VLB5 * Zlipid) + (VNB5 * BETA * Zlipid) + (VWB5 * Zwater)
Zgut5_mf = VLG5_mf * Zlipid + VNG5_mf * BETA * Zlipid + VPG5_mf * GAMMA * Zlipid + VWG5_mf * Zwater
KGB5_mf = Zgut5_mf / Zorg5
KE5_mf = KGB5_mf / WB5 * ED5 * GF5_mf
```

```
    CB5_mf = (CWB_mf * K15_mf * (1 - FPW5) + CSD_mf * K15_mf * FPW5 + KD5_mf * Food5E_mf) / (K25_mf +
        KE5_mf + KG5 + KM5)

    'add check value to Index worksheet
    dic.Add COMPARTMENT5_mf, CB5_mf


'-----------------------------------------
'COMPARTMENT6 (C/O - benthic carnivore/omnivore)
'-----------------------------------------

    WB6 = Worksheets("Output").Cells(41, 9)
    VLB6 = Worksheets("Output").Cells(42, 9)
    VWB6 = Worksheets("Output").Cells(43, 9)
    VNB6 = Worksheets("Output").Cells(45, 9)
    VPB6 = 0
    WBL6 = WB6 * VLB6
    QW6 = 88.3 * WB6 ^ 0.6
    QL6 = QW6 * 0.01

    'Temperature dependent growth
    If TW_mf < 17.5 Then
      KG6 = 0.000502 * WB6 ^ -0.2
    Else
      KG6 = 0.00251 * WB6 ^ -0.2
    End If

    GV6_mf = (1400 * (WB6 ^ 0.65)) / COX_mf
    GD6_mf = 0.022 * WB6 ^ 0.85 * Exp(0.06 * TW_mf)
    DF61 = Worksheets("Diet").Cells(7, 2)
    DF6p = Worksheets("Diet").Cells(7, 3)
    DF62 = Worksheets("Diet").Cells(7, 4)
    DF63 = Worksheets("Diet").Cells(7, 5)
    DF64 = Worksheets("Diet").Cells(7, 6)
    DF65 = Worksheets("Diet").Cells(7, 7)
    eL6 = Worksheets("Output").Cells(46, 9)
    eP6 = Worksheets("Output").Cells(47, 9)
    eN6 = Worksheets("Output").Cells(48, 9)
    eW6 = Worksheets("Output").Cells(49, 9)
    FPW6 = Worksheets("Output").Cells(50, 9)
    Food6A = DF61 * VLBsed + DF6p * VLBpart_DET_mf + DF62 * VLB2 + DF63 * VLB3 + DF64 * VLB4 + DF65 *
        VLB5
    Food6B_mf = DF61 * VNBsed_mf + DF6p * VNBpart_DET_mf + DF62 * VNB2 + DF63 * VNB3 + DF64 * VNB4
        + DF65 * VNB5
    Food6C_mf = DF61 * VWBsed_mf + DF6p * VWBpart_DET_mf + DF62 * VWB2 + DF63 * VWB3 + DF64 *
        VWB4 + DF65 * VWB5
    Food6D_mf = DF61 * VPBsed_mf + DF6p * VPBpart_DET_mf + DF62 * VPB2 + DF63 * VPB3 + DF64 * VPB4
        + DF65 * VPB5
    Food6E_mf = DF61 * CST_mf + DF6p * CFL_mf + DF62 * CB2_mf + DF63 * CB3_mf + DF64 * CB4_mf +
```

```
    DF65 * CB5_mf
  GF6_mf = (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B_mf) + ((1 - eW6) * Food6C_mf) + ((1 - eP6) *
      Food6D_mf)) * GD6_mf
  VLG6_mf = ((1 - eL6) * Food6A) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B_mf) + ((1 - eW6) * Food6C_mf)
      + ((1 - eP6) * Food6D_mf))
  VNG6_mf = ((1 - eN6) * Food6B_mf) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B_mf) + ((1 - eW6) *
      Food6C_mf) + ((1 - eP6) * Food6D_mf))
  VWG6_mf = ((1 - eW6) * Food6C_mf) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B_mf) + ((1 - eW6) *
      Food6C_mf) + ((1 - eP6) * Food6D_mf))
  VPG6_mf = ((1 - eP6) * Food6D_mf) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B_mf) + ((1 - eW6) *
      Food6C_mf) + ((1 - eP6) * Food6D_mf))
  ED6 = 1 / (EDA * KOW + EDB)
  KD6_mf = ED6 * GD6_mf / WB6
  EWW6 = 1 / (1.85 + (155 / KOW))
  K16_mf = EWW6 * GV6_mf / WB6
  KPW6 = (VLB6 * KOW) + (VNB6 * BETA * KOW) + VWB6
  K26_mf = K16_mf / KPW6
  Zorg6 = (VLB6 * Zlipid) + (VNB6 * BETA * Zlipid) + (VWB6 * Zwater)
  Zgut6_mf = VLG6_mf * Zlipid + VNG6_mf * BETA * Zlipid + VPG6_mf * GAMMA * Zlipid + VWG6_mf * Zwater
  KGB6_mf = Zgut6_mf / Zorg6
  KE6_mf = KGB6_mf / WB6 * ED6 * GF6_mf
  CB6_mf = (CWB_mf * K16_mf * (1 - FPW6) + CSD_mf * K16_mf * FPW6 + KD6_mf * Food6E_mf) / (K26_mf +
      KE6_mf + KG6 + KM6)

  'add check value to Index worksheet
  dic.Add COMPARTMENT6_mf, CB6_mf

'------------------------------------------
'COMPARTMENT7 (filter feeding fish)
'------------------------------------------

  WB7 = Worksheets("Output").Cells(41, 10)
  VLB7 = Worksheets("Output").Cells(42, 10)
  VWB7 = Worksheets("Output").Cells(43, 10)
  VNB7 = Worksheets("Output").Cells(45, 10)
  VPB7 = 0
  WBL7 = WB7 * VLB7
  QW7 = 88.3 * WB7 ^ 0.6
  QL7 = QW7 * 0.01

  'Temperature dependent growth
  If TW_mf < 17.5 Then
    KG7 = 0.000502 * WB7 ^ -0.2
  Else
    KG7 = 0.00251 * WB7 ^ -0.2
  End If

  GV7_mf = (1400 * (WB7 ^ 0.65)) / COX_mf
  GD7_mf = 0.022 * WB7 ^ 0.85 * Exp(0.06 * TW_mf)
```

```
DF71 = Worksheets("Diet").Cells(8, 2)
DF7p = Worksheets("Diet").Cells(8, 3)
DF72 = Worksheets("Diet").Cells(8, 4)
DF73 = Worksheets("Diet").Cells(8, 5)
DF74 = Worksheets("Diet").Cells(8, 6)
DF75 = Worksheets("Diet").Cells(8, 7)
DF76 = Worksheets("Diet").Cells(8, 8)
eL7 = Worksheets("Output").Cells(46, 10)
eP7 = Worksheets("Output").Cells(47, 10)
eN7 = Worksheets("Output").Cells(48, 10)
eW7 = Worksheets("Output").Cells(49, 10)
FPW7 = Worksheets("Output").Cells(50, 10)
Food7A = DF71 * VLBsed + DF7p * VLBpart_mf + DF72 * VLB2 + DF73 * VLB3 + DF74 * VLB4 + DF75 *
    VLB5 + DF76 * VLB6
Food7B_mf = DF71 * VNBsed_mf + DF7p * VNBpart_mf + DF72 * VNB2 + DF73 * VNB3 + DF74 * VNB4 +
    DF75 * VNB5 + DF76 * VNB6
Food7C_mf = DF71 * VWBsed_mf + DF7p * VWBpart_mf + DF72 * VWB2 + DF73 * VWB3 + DF74 * VWB4 +
    DF75 * VWB5 + DF76 * VWB6
Food7D_mf = DF71 * VPBsed_mf + DF7p * VPBpart_mf + DF72 * VPB2 + DF73 * VPB3 + DF74 * VPB4 +
    DF75 * VPB5 + DF76 * VPB6
Food7E_mf = DF71 * CST_mf + DF7p * CPART_mf + DF72 * CB2_mf + DF73 * CB3_mf + DF74 * CB4_mf +
    DF75 * CB5_mf + DF76 * CB6_mf
GF7_mf = (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B_mf) + ((1 - eW7) * Food7C_mf) + ((1 - eP7) *
    Food7D_mf)) * GD7_mf
VLG7_mf = ((1 - eL7) * Food7A) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B_mf) + ((1 - eW7) * Food7C_mf)
    + ((1 - eP7) * Food7D_mf))
VNG7_mf = ((1 - eN7) * Food7B_mf) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B_mf) + ((1 - eW7) *
    Food7C_mf) + ((1 - eP7) * Food7D_mf))
VWG7_mf = ((1 - eW7) * Food7C_mf) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B_mf) + ((1 - eW7) *
    Food7C_mf) + ((1 - eP7) * Food7D_mf))
VPG7_mf = ((1 - eP7) * Food7D_mf) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B_mf) + ((1 - eW7) *
    Food7C_mf) + ((1 - eP7) * Food7D_mf))
ED7 = 1 / (EDA * KOW + EDB)
KD7_mf = ED7 * GD7_mf / WB7
EWW7 = 1 / (1.85 + (155 / KOW))
K17_mf = EWW7 * GV7_mf / WB7
KPW7 = (VLB7 * KOW) + (VNB7 * BETA * KOW) + VWB7
K27_mf = K17_mf / KPW7
Zorg7 = (VLB7 * Zlipid) + (VNB7 * BETA * Zlipid) + (VWB7 * Zwater)
Zgut7_mf = VLG7_mf * Zlipid + VNG7_mf * BETA * Zlipid + VPG7_mf * GAMMA * Zlipid + VWG7_mf * Zwater
KGB7_mf = Zgut7_mf / Zorg7
KE7_mf = KGB7_mf / WB7 * ED7 * GF7_mf
CB7_mf = (CWB_mf * K17_mf * (1 - FPW7) + CSD_mf * K17_mf * FPW7 + KD7_mf * Food7E_mf) / (K27_mf +
    KE7_mf + KG7 + KM7)

'add check value to Index worksheet
dic.Add COMPARTMENT7_mf, CB7_mf

'-----------------------------------------------
'COMPARTMENT8 (SFF - small forage fish)
```

```
'-----------------------------------------------

    WB8 = Worksheets("Output").Cells(41, 11)
    VLB8 = Worksheets("Output").Cells(42, 11)
    VWB8 = Worksheets("Output").Cells(43, 11)
    VNB8 = Worksheets("Output").Cells(45, 11)
    VPB8 = 0
    WBL8 = WB8 * VLB8
    QW8 = 88.3 * WB8 ^ 0.6
    QL8 = QW8 * 0.01

    'Temperature dependent growth
    If TW_mf < 17.5 Then
       KG8 = 0.000502 * WB8 ^ -0.2
    Else
       KG8 = 0.00251 * WB8 ^ -0.2
    End If

    GV8_mf = (1400 * (WB8 ^ 0.65)) / COX_mf
    GD8_mf = 0.022 * WB8 ^ 0.85 * Exp(0.06 * TW_mf)
    DF81 = Worksheets("Diet").Cells(9, 2)
    DF8p = Worksheets("Diet").Cells(9, 3)
    DF82 = Worksheets("Diet").Cells(9, 4)
    DF83 = Worksheets("Diet").Cells(9, 5)
    DF84 = Worksheets("Diet").Cells(9, 6)
    DF85 = Worksheets("Diet").Cells(9, 7)
    DF86 = Worksheets("Diet").Cells(9, 8)
    DF87 = Worksheets("Diet").Cells(9, 9)
    eL8 = Worksheets("Output").Cells(46, 11)
    eP8 = Worksheets("Output").Cells(47, 11)
    eN8 = Worksheets("Output").Cells(48, 11)
    eW8 = Worksheets("Output").Cells(49, 11)
    FPW8 = Worksheets("Output").Cells(50, 11)
    Food8A = DF81 * VLBsed + DF8p * VLBpart_DET_mf + DF82 * VLB2 + DF83 * VLB3 + DF84 * VLB4 + DF85 *
        VLB5 + DF86 * VLB6 + DF87 * VLB7
    Food8B_mf = DF81 * VNBsed_mf + DF8p * VNBpart_DET_mf + DF82 * VNB2 + DF83 * VNB3 + DF84 * VNB4
        + DF85 * VNB5 + DF86 * VNB6 + DF87 * VNB7
    Food8C_mf = DF81 * VWBsed_mf + DF8p * VWBpart_DET_mf + DF82 * VWB2 + DF83 * VWB3 + DF84 *
        VWB4 + DF85 * VWB5 + DF86 * VWB6 + DF87 * VWB7
    Food8D_mf = DF81 * VPBsed_mf + DF8p * VPBpart_DET_mf + DF82 * VPB2 + DF83 * VPB3 + DF84 * VPB4
        + DF85 * VPB5 + DF86 * VPB6 + DF87 * VPB7
    Food8E_mf = DF81 * CST_mf + DF8p * CFL_mf + DF82 * CB2_mf + DF83 * CB3_mf + DF84 * CB4_mf +
        DF85 * CB5_mf + DF86 * CB6_mf + DF87 * CB7_mf
    GF8_mf = (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B_mf) + ((1 - eW8) * Food8C_mf) + ((1 - eP8) *
        Food8D_mf)) * GD8_mf
    VLG8_mf = ((1 - eL8) * Food8A) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B_mf) + ((1 - eW8) * Food8C_mf)
        + ((1 - eP8) * Food8D_mf))
    VNG8_mf = ((1 - eN8) * Food8B_mf) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B_mf) + ((1 - eW8) *
        Food8C_mf) + ((1 - eP8) * Food8D_mf))
    VWG8_mf = ((1 - eW8) * Food8C_mf) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B_mf) + ((1 - eW8) *
```

```
        Food8C_mf) + ((1 - eP8) * Food8D_mf))
    VPG8_mf = ((1 - eP8) * Food8D_mf) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B_mf) + ((1 - eW8) *
        Food8C_mf) + ((1 - eP8) * Food8D_mf))
    ED8 = 1 / (EDA * KOW + EDB)
    KD8_mf = ED8 * GD8_mf / WB8
    EWW8 = 1 / (1.85 + (155 / KOW))
    K18_mf = EWW8 * GV8_mf / WB8
    KPW8 = (VLB8 * KOW) + (VNB8 * BETA * KOW) + VWB8
    K28_mf = K18_mf / KPW8
    Zorg8 = (VLB8 * Zlipid) + (VNB8 * BETA * Zlipid) + (VWB8 * Zwater)
    Zgut8_mf = VLG8_mf * Zlipid + VNG8_mf * BETA * Zlipid + VPG8_mf * GAMMA * Zlipid + VWG8_mf * Zwater
    KGB8_mf = Zgut8_mf / Zorg8
    KE8_mf = KGB8_mf / WB8 * ED8 * GF8_mf
    CB8_mf = (CWB_mf * K18_mf * (1 - FPW8) + CSD_mf * K18_mf * FPW8 + KD8_mf * Food8E_mf) / (K28_mf +
        KE8_mf + KG8 + KM8)

    'add check value to Index worksheet
    dic.Add COMPARTMENT8_mf, CB8_mf

'------------------------------
'Riverwide Calculations
'------------------------------



'MODEL CALCULATIONS FOR PHYTOPLANKTON (2)
    VLB2 = Worksheets("Output").Cells(42, 5)
    VWB2 = Worksheets("Output").Cells(43, 5)
    VNB2 = 0
    VPB2 = Worksheets("Output").Cells(44, 5)
    UA = Worksheets("Output").Cells(33, 4)
    UB = Worksheets("Output").Cells(34, 4)
    K12 = 1 / (UA + (UB / KOW))
    KPW2 = (VLB2 * KOW) + (VPB2 * GAMMA * KOW) + VWB2
    K22 = K12 / KPW2
    KG2 = Worksheets("Output").Cells(54, 5)
    FPW2 = Worksheets("Output").Cells(50, 5)
    CB2 = (CWB * K12 * (1 - FPW2)) / (K22 + KG2)
    dic.Add PHYTOPLANKTON, CB2

'MODEL CALCULATIONS FOR ZOOPLANKTON (3)
    WB3 = Worksheets("Output").Cells(41, 6)
    VLB3 = Worksheets("Output").Cells(42, 6)
    VWB3 = Worksheets("Output").Cells(43, 6)
    VNB3 = Worksheets("Output").Cells(45, 6)
    VPB3 = 0
    WBL3 = WB3 * VLB3
    QW3 = 88.3 * WB3 ^ 0.6
    QL3 = QW3 * 0.01
```

```
    GD3 = 0.022 * WB3 ^ 0.85 * Exp(0.06 * TW)

    'Temperature dependent growth
    If TW < 17.5 Then
        KG3 = 0.000502 * WB3 ^ -0.2
    Else
        KG3 = 0.00251 * WB3 ^ -0.2
    End If

    GV3 = (1400 * (WB3 ^ 0.65)) / COX
    DF32 = Worksheets("Diet").Cells(4, 4)
    eL3 = Worksheets("Output").Cells(46, 6)
    eP3 = Worksheets("Output").Cells(47, 6)
    eN3 = Worksheets("Output").Cells(48, 6)
    eW3 = Worksheets("Output").Cells(49, 6)
    Food3A = DF32 * VLB2
    Food3B = DF32 * VNB2
    Food3C = DF32 * VWB2
    Food3D = DF32 * VPB2
    GF3 = (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3) * Food3D)) * GD3
    VLG3 = ((1 - eL3) * Food3A) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    VNG3 = ((1 - eN3) * Food3B) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    VWG3 = ((1 - eW3) * Food3C) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 -
        eP3) * Food3D))
    VPG3 = ((1 - eP3) * Food3D) / (((1 - eL3) * Food3A) + ((1 - eN3) * Food3B) + ((1 - eW3) * Food3C) + ((1 - eP3)
        * Food3D))
    ED3 = 1 / (EDA * KOW + EDB)
    KD3 = ED3 * GD3 / WB3
    EWW3 = 1 / (1.85 + (155 / KOW))
    K13 = EWW3 * GV3 / WB3
    KPW3 = (VLB3 * KOW) + (VNB3 * BETA * KOW) + VWB3
    K23 = K13 / KPW3
    Zorg3 = (VLB3 * Zlipid) + (VNB3 * BETA * Zlipid) + (VWB3 * Zwater)
    Zgut3 = VLG3 * Zlipid + VNG3 * BETA * Zlipid + VPG3 * GAMMA * Zlipid + VWG3 * Zwater
    KGB3 = Zgut3 / Zorg3
    KE3 = KGB3 / WB3 * ED3 * GF3
    FPW3 = Worksheets("Output").Cells(50, 6)
    CB3 = (CWB * K13 * (1 - FPW3) + CB2 * KD3 * DF32) / (K23 + KE3 + KG3 + KM3)
    dic.Add ZOOPLANKTON, CB3

  'MODEL CALCULATIONS FOR COMPARTMENT4 (DEP - benthic deposit feeder)
    WB4 = Worksheets("Output").Cells(41, 7)
    VLB4 = Worksheets("Output").Cells(42, 7)
    VWB4 = Worksheets("Output").Cells(43, 7)
    VNB4 = Worksheets("Output").Cells(45, 7)
    VPB4 = 0
    WBL4 = WB4 * VLB4
```

```
QW4 = 88.3 * WB4 ^ 0.6
QL4 = QW4 * 0.01

'Temperature dependent growth
If TW < 17.5 Then
   KG4 = 0.000502 * WB4 ^ -0.2
Else
   KG4 = 0.00251 * WB4 ^ -0.2
End If

GV4 = (1400 * (WB4 ^ 0.65)) / COX
GD4 = 0.022 * WB4 ^ 0.85 * Exp(0.06 * TW)
DF41 = Worksheets("Diet").Cells(5, 2)
DF4p = Worksheets("Diet").Cells(5, 3)
DF42 = Worksheets("Diet").Cells(5, 4)
DF43 = Worksheets("Diet").Cells(5, 5)
eL4 = Worksheets("Output").Cells(46, 7)
eP4 = Worksheets("Output").Cells(47, 7)
eN4 = Worksheets("Output").Cells(48, 7)
eW4 = Worksheets("Output").Cells(49, 7)
FPW4 = Worksheets("Output").Cells(50, 7)
Food4A = DF41 * VLBsed + DF4p * VLBpart_DET + DF42 * VLB2 + DF43 * VLB3
Food4B = DF41 * VNBsed + DF4p * VNBpart_DET + DF42 * VNB2 + DF43 * VNB3
Food4C = DF41 * VWBsed + DF4p * VWBpart_DET + DF42 * VWB2 + DF43 * VWB3
Food4D = DF41 * VPBsed + DF4p * VPBpart_DET + DF42 * VPB2 + DF43 * VPB3
Food4E = DF41 * CST + DF4p * CFL + DF42 * CB2 + DF43 * CB3
GF4 = (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B) + ((1 - eW4) * Food4C) + ((1 - eP4) * Food4D)) * GD4
VLG4 = ((1 - eL4) * Food4A) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B) + ((1 - eW4) * Food4C) + ((1 - eP4)
    * Food4D))
VNG4 = ((1 - eN4) * Food4B) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B) + ((1 - eW4) * Food4C) + ((1 - eP4)
    * Food4D))
VWG4 = ((1 - eW4) * Food4C) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B) + ((1 - eW4) * Food4C) + ((1 -
    eP4) * Food4D))
VPG4 = ((1 - eP4) * Food4D) / (((1 - eL4) * Food4A) + ((1 - eN4) * Food4B) + ((1 - eW4) * Food4C) + ((1 - eP4)
    * Food4D))
ED4 = 1 / (EDA * KOW + EDB)
KD4 = ED4 * GD4 / WB4
EWW4 = 1 / (1.85 + (155 / KOW))
K14 = EWW4 * GV4 / WB4
KPW4 = (VLB4 * KOW) + (VNB4 * BETA * KOW) + VWB4
K24 = K14 / KPW4
Zorg4 = (VLB4 * Zlipid) + (VNB4 * BETA * Zlipid) + (VWB4 * Zwater)
Zgut4 = VLG4 * Zlipid + VNG4 * BETA * Zlipid + VPG4 * GAMMA * Zlipid + VWG4 * Zwater
KGB4 = Zgut4 / Zorg4
KE4 = KGB4 / WB4 * ED4 * GF4
CB4 = (CWB * K14 * (1 - FPW4) + K14 * FPW4 * CSD + KD4 * Food4E) / (K24 + KE4 + KG4 + KM4)
dic.Add COMPARTMENT4, CB4

'MODEL CALCULATIONS FOR COMPARTMENT5 (DET - benthic detritivore)
```

```
WB5 = Worksheets("Output").Cells(41, 8)
VLB5 = Worksheets("Output").Cells(42, 8)
VWB5 = Worksheets("Output").Cells(43, 8)
VNB5 = Worksheets("Output").Cells(45, 8)
VPB5 = 0
WBL5 = WB5 * VLB5
QW5 = 88.3 * WB5 ^ 0.6
QL5 = QW5 * 0.01

'Temperature dependent growth
If TW < 17.5 Then
   KG5 = 0.000502 * WB5 ^ -0.2
Else
   KG5 = 0.00251 * WB5 ^ -0.2
End If

GV5 = (1400 * (WB5 ^ 0.65)) / COX
GD5 = 0.022 * WB5 ^ 0.85 * Exp(0.06 * TW)
DF51 = Worksheets("Diet").Cells(6, 2)
DF5p = Worksheets("Diet").Cells(6, 3)
DF52 = Worksheets("Diet").Cells(6, 4)
DF53 = Worksheets("Diet").Cells(6, 5)
DF54 = Worksheets("Diet").Cells(6, 6)
eL5 = Worksheets("Output").Cells(46, 8)
eP5 = Worksheets("Output").Cells(47, 8)
eN5 = Worksheets("Output").Cells(48, 8)
eW5 = Worksheets("Output").Cells(49, 8)
FPW5 = Worksheets("Output").Cells(50, 8)
Food5A = DF51 * VLBsed + DF5p * VLBpart_DET + DF52 * VLB2 + DF53 * VLB3 + DF54 * VLB4
Food5B = DF51 * VNBsed + DF5p * VNBpart_DET + DF52 * VNB2 + DF53 * VNB3 + DF54 * VNB4
Food5C = DF51 * VWBsed + DF5p * VWBpart_DET + DF52 * VWB2 + DF53 * VWB3 + DF54 * VWB4
Food5D = DF51 * VPBsed + DF5p * VPBpart_DET + DF52 * VPB2 + DF53 * VPB3 + DF54 * VPB4
Food5E = DF51 * CST + DF5p * CFL + DF52 * CB2 + DF53 * CB3 + DF54 * CB4
GF5 = (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B) + ((1 - eW5) * Food5C) + ((1 - eP5) * Food5D)) * GD5
VLG5 = ((1 - eL5) * Food5A) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B) + ((1 - eW5) * Food5C) + ((1 - eP5)
     * Food5D))
VNG5 = ((1 - eN5) * Food5B) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B) + ((1 - eW5) * Food5C) + ((1 - eP5)
     * Food5D))
VWG5 = ((1 - eW5) * Food5C) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B) + ((1 - eW5) * Food5C) + ((1 -
     eP5) * Food5D))
VPG5 = ((1 - eP5) * Food5D) / (((1 - eL5) * Food5A) + ((1 - eN5) * Food5B) + ((1 - eW5) * Food5C) + ((1 - eP5)
     * Food5D))
ED5 = 1 / (EDA * KOW + EDB)
KD5 = ED5 * GD5 / WB5
EWW5 = 1 / (1.85 + (155 / KOW))
K15 = EWW5 * GV5 / WB5
KPW5 = (VLB5 * KOW) + (VNB5 * BETA * KOW) + VWB5
K25 = K15 / KPW5
Zorg5 = (VLB5 * Zlipid) + (VNB5 * BETA * Zlipid) + (VWB5 * Zwater)
```

```
  Zgut5 = VLG5 * Zlipid + VNG5 * BETA * Zlipid + VPG5 * GAMMA * Zlipid + VWG5 * Zwater
  KGB5 = Zgut5 / Zorg5
  KE5 = KGB5 / WB5 * ED5 * GF5
  CB5 = (CWB * K15 * (1 - FPW5) + CSD * K15 * FPW5 + KD5 * Food5E) / (K25 + KE5 + KG5 + KM5)
  dic.Add COMPARTMENT5, CB5

'MODEL CALCULATIONS FOR COMPARTMENT6 (C/O – benthic carnivore/omnivore)
  WB6 = Worksheets("Output").Cells(41, 9)
  VLB6 = Worksheets("Output").Cells(42, 9)
  VWB6 = Worksheets("Output").Cells(43, 9)
  VNB6 = Worksheets("Output").Cells(45, 9)
  VPB6 = 0
  WBL6 = WB6 * VLB6
  QW6 = 88.3 * WB6 ^ 0.6
  QL6 = QW6 * 0.01

  'Temperature dependent growth
  If TW < 17.5 Then
    KG6 = 0.000502 * WB6 ^ -0.2
  Else
    KG6 = 0.00251 * WB6 ^ -0.2
  End If

  GV6 = (1400 * (WB6 ^ 0.65)) / COX
  GD6 = 0.022 * WB6 ^ 0.85 * Exp(0.06 * TW)
  DF61 = Worksheets("Diet").Cells(7, 2)
  DF6p = Worksheets("Diet").Cells(7, 3)
  DF62 = Worksheets("Diet").Cells(7, 4)
  DF63 = Worksheets("Diet").Cells(7, 5)
  DF64 = Worksheets("Diet").Cells(7, 6)
  DF65 = Worksheets("Diet").Cells(7, 7)
  eL6 = Worksheets("Output").Cells(46, 9)
  eP6 = Worksheets("Output").Cells(47, 9)
  eN6 = Worksheets("Output").Cells(48, 9)
  eW6 = Worksheets("Output").Cells(49, 9)
  FPW6 = Worksheets("Output").Cells(50, 9)
  Food6A = DF61 * VLBsed + DF6p * VLBpart_DET + DF62 * VLB2 + DF63 * VLB3 + DF64 * VLB4 + DF65 *
      VLB5
  Food6B = DF61 * VNBsed + DF6p * VNBpart_DET + DF62 * VNB2 + DF63 * VNB3 + DF64 * VNB4 + DF65 *
      VNB5
  Food6C = DF61 * VWBsed + DF6p * VWBpart_DET + DF62 * VWB2 + DF63 * VWB3 + DF64 * VWB4 + DF65
      * VWB5
  Food6D = DF61 * VPBsed + DF6p * VPBpart_DET + DF62 * VPB2 + DF63 * VPB3 + DF64 * VPB4 + DF65 *
      VPB5
  Food6E = DF61 * CST + DF6p * CFL + DF62 * CB2 + DF63 * CB3 + DF64 * CB4 + DF65 * CB5
  GF6 = (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B) + ((1 - eW6) * Food6C) + ((1 - eP6) * Food6D)) * GD6
  VLG6 = ((1 - eL6) * Food6A) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B) + ((1 - eW6) * Food6C) + ((1 - eP6)
      * Food6D))
  VNG6 = ((1 - eN6) * Food6B) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B) + ((1 - eW6) * Food6C) + ((1 - eP6)
```

```
        * Food6D))
    VWG6 = ((1 - eW6) * Food6C) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B) + ((1 - eW6) * Food6C) + ((1 -
        eP6) * Food6D))
    VPG6 = ((1 - eP6) * Food6D) / (((1 - eL6) * Food6A) + ((1 - eN6) * Food6B) + ((1 - eW6) * Food6C) + ((1 - eP6)
        * Food6D))
    ED6 = 1 / (EDA * KOW + EDB)
    KD6 = ED6 * GD6 / WB6
    EWW6 = 1 / (1.85 + (155 / KOW))
    K16 = EWW6 * GV6 / WB6
    KPW6 = (VLB6 * KOW) + (VNB6 * BETA * KOW) + VWB6
    K26 = K16 / KPW6
    Zorg6 = (VLB6 * Zlipid) + (VNB6 * BETA * Zlipid) + (VWB6 * Zwater)
    Zgut6 = VLG6 * Zlipid + VNG6 * BETA * Zlipid + VPG6 * GAMMA * Zlipid + VWG6 * Zwater
    KGB6 = Zgut6 / Zorg6
    KE6 = KGB6 / WB6 * ED6 * GF6
    CB6 = (CWB * K16 * (1 - FPW6) + CSD * K16 * FPW6 + KD6 * Food6E) / (K26 + KE6 + KG6 + KM6)
    dic.Add COMPARTMENT6, CB6

'MODEL CALCULATIONS FOR COMPARTMENT7 (FFF - filter feeding fish)
    WB7 = Worksheets("Output").Cells(41, 10)
    VLB7 = Worksheets("Output").Cells(42, 10)
    VWB7 = Worksheets("Output").Cells(43, 10)
    VNB7 = Worksheets("Output").Cells(45, 10)
    VPB7 = 0
    WBL7 = WB7 * VLB7
    QW7 = 88.3 * WB7 ^ 0.6
    QL7 = QW7 * 0.01

    'Temperature dependent growth
    If TW < 17.5 Then
        KG7 = 0.000502 * WB7 ^ -0.2
    Else
        KG7 = 0.00251 * WB7 ^ -0.2
    End If

    GV7 = (1400 * (WB7 ^ 0.65)) / COX
    GD7 = 0.022 * WB7 ^ 0.85 * Exp(0.06 * TW)
    DF71 = Worksheets("Diet").Cells(8, 2)
    DF7p = Worksheets("Diet").Cells(8, 3)
    DF72 = Worksheets("Diet").Cells(8, 4)
    DF73 = Worksheets("Diet").Cells(8, 5)
    DF74 = Worksheets("Diet").Cells(8, 6)
    DF75 = Worksheets("Diet").Cells(8, 7)
    DF76 = Worksheets("Diet").Cells(8, 8)
    eL7 = Worksheets("Output").Cells(46, 10)
    eP7 = Worksheets("Output").Cells(47, 10)
    eN7 = Worksheets("Output").Cells(48, 10)
    eW7 = Worksheets("Output").Cells(49, 10)
```

```
FPW7 = Worksheets("Output").Cells(50, 10)
Food7A = DF71 * VLBsed + DF7p * VLBpart + DF72 * VLB2 + DF73 * VLB3 + DF74 * VLB4 + DF75 * VLB5 +
    DF76 * VLB6
Food7B = DF71 * VNBsed + DF7p * VNBpart + DF72 * VNB2 + DF73 * VNB3 + DF74 * VNB4 + DF75 * VNB5
    + DF76 * VNB6
Food7C = DF71 * VWBsed + DF7p * VWBpart + DF72 * VWB2 + DF73 * VWB3 + DF74 * VWB4 + DF75 *
    VWB5 + DF76 * VWB6
Food7D = DF71 * VPBsed + DF7p * VPBpart + DF72 * VPB2 + DF73 * VPB3 + DF74 * VPB4 + DF75 * VPB5
    + DF76 * VPB6
Food7E = DF71 * CST + DF7p * CPART + DF72 * CB2 + DF73 * CB3 + DF74 * CB4 + DF75 * CB5 + DF76 *
    CB6
GF7 = (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B) + ((1 - eW7) * Food7C) + ((1 - eP7) * Food7D)) * GD7
VLG7 = ((1 - eL7) * Food7A) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B) + ((1 - eW7) * Food7C) + ((1 - eP7)
    * Food7D))
VNG7 = ((1 - eN7) * Food7B) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B) + ((1 - eW7) * Food7C) + ((1 - eP7)
    * Food7D))
VWG7 = ((1 - eW7) * Food7C) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B) + ((1 - eW7) * Food7C) + ((1 -
    eP7) * Food7D))
VPG7 = ((1 - eP7) * Food7D) / (((1 - eL7) * Food7A) + ((1 - eN7) * Food7B) + ((1 - eW7) * Food7C) + ((1 - eP7)
    * Food7D))
ED7 = 1 / (EDA * KOW + EDB)
KD7 = ED7 * GD7 / WB7
EWW7 = 1 / (1.85 + (155 / KOW))
K17 = EWW7 * GV7 / WB7
KPW7 = (VLB7 * KOW) + (VNB7 * BETA * KOW) + VWB7
K27 = K17 / KPW7
Zorg7 = (VLB7 * Zlipid) + (VNB7 * BETA * Zlipid) + (VWB7 * Zwater)
Zgut7 = VLG7 * Zlipid + VNG7 * BETA * Zlipid + VPG7 * GAMMA * Zlipid + VWG7 * Zwater
KGB7 = Zgut7 / Zorg7
KE7 = KGB7 / WB7 * ED7 * GF7
CB7 = (CWB * K17 * (1 - FPW7) + CSD * K17 * FPW7 + KD7 * Food7E) / (K27 + KE7 + KG7 + KM7)
dic.Add COMPARTMENT7, CB7

'MODEL CALCULATIONS FOR COMPARTMENT8 (SFF - small forage fish)
WB8 = Worksheets("Output").Cells(41, 11)
VLB8 = Worksheets("Output").Cells(42, 11)
VWB8 = Worksheets("Output").Cells(43, 11)
VNB8 = Worksheets("Output").Cells(45, 11)
VPB8 = 0
WBL8 = WB8 * VLB8
QW8 = 88.3 * WB8 ^ 0.6
QL8 = QW8 * 0.01

'Temperature dependent growth
If TW < 17.5 Then
   KG8 = 0.000502 * WB8 ^ -0.2
Else
   KG8 = 0.00251 * WB8 ^ -0.2
End If
```

```
   GV8 = (1400 * (WB8 ^ 0.65)) / COX
   GD8 = 0.022 * WB8 ^ 0.85 * Exp(0.06 * TW)
   DF81 = Worksheets("Diet").Cells(9, 2)
   DF8p = Worksheets("Diet").Cells(9, 3)
   DF82 = Worksheets("Diet").Cells(9, 4)
   DF83 = Worksheets("Diet").Cells(9, 5)
   DF84 = Worksheets("Diet").Cells(9, 6)
   DF85 = Worksheets("Diet").Cells(9, 7)
   DF86 = Worksheets("Diet").Cells(9, 8)
   DF87 = Worksheets("Diet").Cells(9, 9)
   eL8 = Worksheets("Output").Cells(46, 11)
   eP8 = Worksheets("Output").Cells(47, 11)
   eN8 = Worksheets("Output").Cells(48, 11)
   eW8 = Worksheets("Output").Cells(49, 11)
   FPW8 = Worksheets("Output").Cells(50, 11)
   Food8A = DF81 * VLBsed + DF8p * VLBpart_DET + DF82 * VLB2 + DF83 * VLB3 + DF84 * VLB4 + DF85 *
        VLB5 + DF86 * VLB6 + DF87 * VLB7
   Food8B = DF81 * VNBsed + DF8p * VNBpart_DET + DF82 * VNB2 + DF83 * VNB3 + DF84 * VNB4 + DF85 *
        VNB5 + DF86 * VNB6 + DF87 * VNB7
   Food8C = DF81 * VWBsed + DF8p * VWBpart_DET + DF82 * VWB2 + DF83 * VWB3 + DF84 * VWB4 + DF85
        * VWB5 + DF86 * VWB6 + DF87 * VWB7
   Food8D = DF81 * VPBsed + DF8p * VPBpart_DET + DF82 * VPB2 + DF83 * VPB3 + DF84 * VPB4 + DF85 *
        VPB5 + DF86 * VPB6 + DF87 * VPB7
   Food8E = DF81 * CST + DF8p * CFL + DF82 * CB2 + DF83 * CB3 + DF84 * CB4 + DF85 * CB5 + DF86 * CB6
        + DF87 * CB7
   GF8 = (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8) * Food8D)) * GD8
   VLG8 = ((1 - eL8) * Food8A) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8)
        * Food8D))
   VNG8 = ((1 - eN8) * Food8B) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8)
        * Food8D))
   VWG8 = ((1 - eW8) * Food8C) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 -
        eP8) * Food8D))
   VPG8 = ((1 - eP8) * Food8D) / (((1 - eL8) * Food8A) + ((1 - eN8) * Food8B) + ((1 - eW8) * Food8C) + ((1 - eP8)
        * Food8D))
   ED8 = 1 / (EDA * KOW + EDB)
   KD8 = ED8 * GD8 / WB8
   EWW8 = 1 / (1.85 + (155 / KOW))
   K18 = EWW8 * GV8 / WB8
   KPW8 = (VLB8 * KOW) + (VNB8 * BETA * KOW) + VWB8
   K28 = K18 / KPW8
   Zorg8 = (VLB8 * Zlipid) + (VNB8 * BETA * Zlipid) + (VWB8 * Zwater)
   Zgut8 = VLG8 * Zlipid + VNG8 * BETA * Zlipid + VPG8 * GAMMA * Zlipid + VWG8 * Zwater
   KGB8 = Zgut8 / Zorg8
   KE8 = KGB8 / WB8 * ED8 * GF8
   CB8 = (CWB * K18 * (1 - FPW8) + CSD * K18 * FPW8 + KD8 * Food8E) / (K28 + KE8 + KG8 + KM8)
   dic.Add COMPARTMENT8, CB8

'MODEL CALCULATIONS FOR COMPARTMENT9 (CAR - carp)
   WB9 = Worksheets("Output").Cells(41, 12)
   VLB9 = Worksheets("Output").Cells(42, 12)
```

```
VWB9 = Worksheets("Output").Cells(43, 12)
VNB9 = Worksheets("Output").Cells(45, 12)
VPB9 = 0
WBL9 = WB9 * VLB9
QW9 = 88.3 * WB9 ^ 0.6
QL9 = QW9 * 0.01

'Temperature dependent growth
If TW < 17.5 Then
   KG9 = 0.000502 * WB9 ^ -0.2
Else
   KG9 = 0.00251 * WB9 ^ -0.2
End If

GV9 = (1400 * (WB9 ^ 0.65)) / COX
GD9 = 0.022 * WB9 ^ 0.85 * Exp(0.06 * TW)
DF91 = Worksheets("Diet").Cells(10, 2)
DF9p = Worksheets("Diet").Cells(10, 3)
DF92 = Worksheets("Diet").Cells(10, 4)
DF93 = Worksheets("Diet").Cells(10, 5)
DF94 = Worksheets("Diet").Cells(10, 6)
DF95 = Worksheets("Diet").Cells(10, 7)
DF96 = Worksheets("Diet").Cells(10, 8)
DF97 = Worksheets("Diet").Cells(10, 9)
DF98 = Worksheets("Diet").Cells(10, 10)
eL9 = Worksheets("Output").Cells(46, 12)
eP9 = Worksheets("Output").Cells(47, 12)
eN9 = Worksheets("Output").Cells(48, 12)
eW9 = Worksheets("Output").Cells(49, 12)
FPW9 = Worksheets("Output").Cells(50, 12)
Food9A = DF91 * VLBsed + DF9p * VLBpart_DET + DF92 * VLB2 + DF93 * VLB3 + DF94 * VLB4 + DF95 *
    VLB5 + DF96 * VLB6 + DF97 * VLB7 + DF98 * VLB8
Food9B = DF91 * VNBsed + DF9p * VNBpart_DET + DF92 * VNB2 + DF93 * VNB3 + DF94 * VNB4 + DF95 *
    VNB5 + DF96 * VNB6 + DF97 * VNB7 + DF98 * VNB8
Food9C = DF91 * VWBsed + DF9p * VWBpart_DET + DF92 * VWB2 + DF93 * VWB3 + DF94 * VWB4 + DF95
    * VWB5 + DF96 * VWB6 + DF97 * VWB7 + DF98 * VWB8
Food9D = DF91 * VPBsed + DF9p * VPBpart_DET + DF92 * VPB2 + DF93 * VPB3 + DF94 * VPB4 + DF95 *
    VPB5 + DF96 * VPB6 + DF97 * VPB7 + DF98 * VPB8
Food9E = DF91 * CST + DF9p * CFL + DF92 * CB2 + DF93 * CB3 + DF94 * CB4 + DF95 * CB5 + DF96 * CB6
    + DF97 * CB7 + DF98 * CB8_mf
GF9 = (((1 - eL9) * Food9A) + ((1 - eN9) * Food9B) + ((1 - eW9) * Food9C) + ((1 - eP9) * Food9D)) * GD9
VLG9 = ((1 - eL9) * Food9A) / (((1 - eL9) * Food9A) + ((1 - eN9) * Food9B) + ((1 - eW9) * Food9C) + ((1 - eP9)
    * Food9D))
VNG9 = ((1 - eN9) * Food9B) / (((1 - eL9) * Food9A) + ((1 - eN9) * Food9B) + ((1 - eW9) * Food9C) + ((1 - eP9)
    * Food9D))
VWG9 = ((1 - eW9) * Food9C) / (((1 - eL9) * Food9A) + ((1 - eN9) * Food9B) + ((1 - eW9) * Food9C) + ((1 -
    eP9) * Food9D))
VPG9 = ((1 - eP9) * Food9D) / (((1 - eL9) * Food9A) + ((1 - eN9) * Food9B) + ((1 - eW9) * Food9C) + ((1 - eP9)
    * Food9D))
ED9 = 1 / (EDA * KOW + EDB)
```

```
    KD9 = ED9 * GD9 / WB9
    EWW9 = 1 / (1.85 + (155 / KOW))
    K19 = EWW9 * GV9 / WB9
    KPW9 = (VLB9 * KOW) + (VNB9 * BETA * KOW) + VWB9
    K29 = K19 / KPW9
    Zorg9 = (VLB9 * Zlipid) + (VNB9 * BETA * Zlipid) + (VWB9 * Zwater)
    Zgut9 = VLG9 * Zlipid + VNG9 * BETA * Zlipid + VPG9 * GAMMA * Zlipid + VWG9 * Zwater
    KGB9 = Zgut9 / Zorg9
    KE9 = KGB9 / WB9 * ED9 * GF9
    CB9 = (CPV * (CWB * K19 * (1 - FPW9) + CSD * K19 * FPW9) + KD9 * Food9E) / (K29 + KE9 + KG9 + KM9)
    dic.Add COMPARTMENT9, CB9

'MODEL CALCULATIONS FOR COMPARTMENT10 (CAT - catfish)
    WB10 = Worksheets("Output").Cells(41, 13)
    VLB10 = Worksheets("Output").Cells(42, 13)
    VWB10 = Worksheets("Output").Cells(43, 13)
    VNB10 = Worksheets("Output").Cells(45, 13)
    VPB10 = 0
    WBL10 = WB10 * VLB10
    QW10 = 88.3 * WB10 ^ 0.6
    QL10 = QW10 * 0.01

    'Temperature dependent growth
    If TW < 17.5 Then
      KG10 = 0.000502 * WB10 ^ -0.2
    Else
      KG10 = 0.00251 * WB10 ^ -0.2
    End If

    GV10 = (1400 * (WB10 ^ 0.65)) / COX
    GD10 = 0.022 * WB10 ^ 0.85 * Exp(0.06 * TW)
    DF101 = Worksheets("Diet").Cells(11, 2)
    DF10p = Worksheets("Diet").Cells(11, 3)
    DF102 = Worksheets("Diet").Cells(11, 4)
    DF103 = Worksheets("Diet").Cells(11, 5)
    DF104 = Worksheets("Diet").Cells(11, 6)
    DF105 = Worksheets("Diet").Cells(11, 7)
    DF106 = Worksheets("Diet").Cells(11, 8)
    DF107 = Worksheets("Diet").Cells(11, 9)
    DF108 = Worksheets("Diet").Cells(11, 10)
    DF109 = Worksheets("Diet").Cells(11, 11)
    eL10 = Worksheets("Output").Cells(46, 13)
    eP10 = Worksheets("Output").Cells(47, 13)
    eN10 = Worksheets("Output").Cells(48, 13)
    eW10 = Worksheets("Output").Cells(49, 13)
    FPW10 = Worksheets("Output").Cells(50, 13)
    Food10A = DF101 * VLBsed + DF10p * VLBpart_DET + DF102 * VLB2 + DF103 * VLB3 + DF104 * VLB4 +
        DF105 * VLB5 + DF106 * VLB6 + DF107 * VLB7 + DF108 * VLB8 + DF109 * VLB9
```

```
  Food10B = DF101 * VNBsed + DF10p * VNBpart_DET + DF102 * VNB2 + DF103 * VNB3 + DF104 * VNB4 +
      DF105 * VNB5 + DF106 * VNB6 + DF107 * VNB7 + DF108 * VNB8 + DF109 * VNB9
  Food10C = DF101 * VWBsed + DF10p * VWBpart_DET + DF102 * VWB2 + DF103 * VWB3 + DF104 * VWB4
      + DF105 * VWB5 + DF106 * VWB6 + DF107 * VWB7 + DF108 * VWB8 + DF109 * VWB9
  Food10D = DF101 * VPBsed + DF10p * VPBpart_DET + DF102 * VPB2 + DF103 * VPB3 + DF104 * VPB4 +
      DF105 * VPB5 + DF106 * VPB6 + DF107 * VPB7 + DF108 * VPB8 + DF109 * VPB9
  Food10E = DF101 * CST + DF10p * CFL + DF102 * CB2 + DF103 * CB3 + DF104 * CB4 + DF105 * CB5 +
      DF106 * CB6 + DF107 * CB7 + DF108 * CB8_mf + DF109 * CB9
  GF10 = (((1 - eL10) * Food10A) + ((1 - eN10) * Food10B) + ((1 - eW10) * Food10C) + ((1 - eP10) * Food10D))
      * GD10
  VLG10 = ((1 - eL10) * Food10A) / (((1 - eL10) * Food10A) + ((1 - eN10) * Food10B) + ((1 - eW10) * Food10C)
      + ((1 - eP10) * Food10D))
  VNG10 = ((1 - eN10) * Food10B) / (((1 - eL10) * Food10A) + ((1 - eN10) * Food10B) + ((1 - eW10) * Food10C)
      + ((1 - eP10) * Food10D))
  VWG10 = ((1 - eW10) * Food10C) / (((1 - eL10) * Food10A) + ((1 - eN10) * Food10B) + ((1 - eW10) *
      Food10C) + ((1 - eP10) * Food10D))
  VPG10 = ((1 - eP10) * Food10D) / (((1 - eL10) * Food10A) + ((1 - eN10) * Food10B) + ((1 - eW10) * Food10C)
      + ((1 - eP10) * Food10D))
  ED10 = 1 / (EDA * KOW + EDB)
  KD10 = ED10 * GD10 / WB10
  EWW10 = 1 / (1.85 + (155 / KOW))
  K110 = EWW10 * GV10 / WB10
  KPW10 = (VLB10 * KOW) + (VNB10 * BETA * KOW) + VWB10
  K210 = K110 / KPW10
  Zorg10 = (VLB10 * Zlipid) + (VNB10 * BETA * Zlipid) + (VWB10 * Zwater)
  Zgut10 = VLG10 * Zlipid + VNG10 * BETA * Zlipid + VPG10 * GAMMA * Zlipid + VWG10 * Zwater
  KGB10 = Zgut10 / Zorg10
  KE10 = KGB10 / WB10 * ED10 * GF10
  CB10 = (CWB * K110 * (1 - FPW10) + CSD * K110 * FPW10 + KD10 * Food10E) / (K210 + KE10 + KG10 +
      KM10)
  dic.Add COMPARTMENT10, CB10

'MODEL CALCULATIONS FOR COMPARTMENT11 (WP - white perch)
  WB11 = Worksheets("Output").Cells(41, 14)
  VLB11 = Worksheets("Output").Cells(42, 14)
  VWB11 = Worksheets("Output").Cells(43, 14)
  VNB11 = Worksheets("Output").Cells(45, 14)
  VPB11 = 0
  WBL11 = WB11 * VLB11
  QW11 = 88.3 * WB11 ^ 0.6
  QL11 = QW11 * 0.01

  'Temperature dependent growth
  If TW < 17.5 Then
    KG11 = 0.000502 * WB11 ^ -0.2
  Else
    KG11 = 0.00251 * WB11 ^ -0.2
  End If

  GV11 = (1400 * (WB11 ^ 0.65)) / COX
```

```
GD11 = 0.022 * WB11 ^ 0.85 * Exp(0.06 * TW)
DF111 = Worksheets("Diet").Cells(12, 2)
DF11p = Worksheets("Diet").Cells(12, 3)
DF112 = Worksheets("Diet").Cells(12, 4)
DF113 = Worksheets("Diet").Cells(12, 5)
DF114 = Worksheets("Diet").Cells(12, 6)
DF115 = Worksheets("Diet").Cells(12, 7)
DF116 = Worksheets("Diet").Cells(12, 8)
DF117 = Worksheets("Diet").Cells(12, 9)
DF118 = Worksheets("Diet").Cells(12, 10)
DF119 = Worksheets("Diet").Cells(12, 11)
DF1110 = Worksheets("Diet").Cells(12, 12)
eL11 = Worksheets("Output").Cells(46, 14)
eP11 = Worksheets("Output").Cells(47, 14)
eN11 = Worksheets("Output").Cells(48, 14)
eW11 = Worksheets("Output").Cells(49, 14)
FPW11 = Worksheets("Output").Cells(50, 14)
Food11A = DF111 * VLBsed + DF11p * VLBpart_DET + DF112 * VLB2 + DF113 * VLB3 + DF114 * VLB4 +
    DF115 * VLB5 + DF116 * VLB6 + DF117 * VLB7 + DF118 * VLB8 + DF119 * VLB9 + DF1110 * VLB10
Food11B = DF111 * VNBsed + DF11p * VNBpart_DET + DF112 * VNB2 + DF113 * VNB3 + DF114 * VNB4 +
    DF115 * VNB5 + DF116 * VNB6 + DF117 * VNB7 + DF118 * VNB8 + DF119 * VNB9 + DF1110 * VNB10
Food11C = DF111 * VWBsed + DF11p * VWBpart_DET + DF112 * VWB2 + DF113 * VWB3 + DF114 * VWB4
    + DF115 * VWB5 + DF116 * VWB6 + DF117 * VWB7 + DF118 * VWB8 + DF119 * VWB9 + DF1110 *
    VWB10
Food11D = DF111 * VPBsed + DF11p * VPBpart_DET + DF112 * VPB2 + DF113 * VPB3 + DF114 * VPB4 +
    DF115 * VPB5 + DF116 * VPB6 + DF117 * VPB7 + DF118 * VPB8 + DF119 * VPB9 + DF1110 * VPB10
Food11E = DF111 * CST + DF11p * CFL + DF112 * CB2 + DF113 * CB3 + DF114 * CB4 + DF115 * CB5 +
    DF116 * CB6 + DF117 * CB7 + DF118 * CB8_mf + DF119 * CB9 + DF1110 * CB10
GF11 = (((1 - eL11) * Food11A) + ((1 - eN11) * Food11B) + ((1 - eW11) * Food11C) + ((1 - eP11) * Food11D))
    * GD11
VLG11 = ((1 - eL11) * Food11A) / (((1 - eL11) * Food11A) + ((1 - eN11) * Food11B) + ((1 - eW11) * Food11C)
    + ((1 - eP11) * Food11D))
VNG11 = ((1 - eN11) * Food11B) / (((1 - eL11) * Food11A) + ((1 - eN11) * Food11B) + ((1 - eW11) * Food11C)
    + ((1 - eP11) * Food11D))
VWG11 = ((1 - eW11) * Food11C) / (((1 - eL11) * Food11A) + ((1 - eN11) * Food11B) + ((1 - eW11) *
    Food11C) + ((1 - eP11) * Food11D))
VPG11 = ((1 - eP11) * Food11D) / (((1 - eL11) * Food11A) + ((1 - eN11) * Food11B) + ((1 - eW11) * Food11C)
    + ((1 - eP11) * Food11D))
ED11 = 1 / (EDA * KOW + EDB)
KD11 = ED11 * GD11 / WB11
EWW11 = 1 / (1.85 + (155 / KOW))
K111 = EWW11 * GV11 / WB11
KPW11 = (VLB11 * KOW) + (VNB11 * BETA * KOW) + VWB11
K211 = K111 / KPW11
Zorg11 = (VLB11 * Zlipid) + (VNB11 * BETA * Zlipid) + (VWB11 * Zwater)
Zgut11 = VLG11 * Zlipid + VNG11 * BETA * Zlipid + VPG11 * GAMMA * Zlipid + VWG11 * Zwater
KGB11 = Zgut11 / Zorg11
KE11 = KGB11 / WB11 * ED11 * GF11
CB11 = (CWB * K111 * (1 - FPW11) + CSD * K111 * FPW11 + KD11 * Food11E) / (K211 + KE11 + KG11 +
    KM11)
dic.Add COMPARTMENT11, CB11
```

```
'MODEL CALCULATIONS FOR COMPARTMENT12 (AE - american eel)
  WB12 = Worksheets("Output").Cells(41, 15)
  VLB12 = Worksheets("Output").Cells(42, 15)
  VWB12 = Worksheets("Output").Cells(43, 15)
  VNB12 = Worksheets("Output").Cells(45, 15)
  VPB12 = 0
  WBL12 = WB12 * VLB12
  QW12 = 88.3 * WB12 ^ 0.6
  QL12 = QW12 * 0.01

  'Temperature dependent growth
  If TW < 17.5 Then
    KG12 = 0.000502 * WB12 ^ -0.2
  Else
    KG12 = 0.00251 * WB12 ^ -0.2
  End If

  GV12 = (1400 * (WB12 ^ 0.65)) / COX
  GD12 = 0.022 * WB12 ^ 0.85 * Exp(0.06 * TW)
  DF121 = Worksheets("Diet").Cells(13, 2)
  DF12p = Worksheets("Diet").Cells(13, 3)
  DF122 = Worksheets("Diet").Cells(13, 4)
  DF123 = Worksheets("Diet").Cells(13, 5)
  DF124 = Worksheets("Diet").Cells(13, 6)
  DF125 = Worksheets("Diet").Cells(13, 7)
  DF126 = Worksheets("Diet").Cells(13, 8)
  DF127 = Worksheets("Diet").Cells(13, 9)
  DF128 = Worksheets("Diet").Cells(13, 10)
  DF129 = Worksheets("Diet").Cells(13, 11)
  DF1210 = Worksheets("Diet").Cells(13, 12)
  DF1211 = Worksheets("Diet").Cells(13, 13)
  eL12 = Worksheets("Output").Cells(46, 15)
  eP12 = Worksheets("Output").Cells(47, 15)
  eN12 = Worksheets("Output").Cells(48, 15)
  eW12 = Worksheets("Output").Cells(49, 15)
  FPW12 = Worksheets("Output").Cells(50, 15)
  Food12A = DF121 * VLBsed + DF12p * VLBpart_DET + DF122 * VLB2 + DF123 * VLB3 + DF124 * VLB4 +
      DF125 * VLB5 + DF126 * VLB6 + DF127 * VLB7 + DF128 * VLB8 + DF129 * VLB9 + DF1210 * VLB10 +
      DF1211 * VLB11
  Food12B = DF121 * VNBsed + DF12p * VNBpart_DET + DF122 * VNB2 + DF123 * VNB3 + DF124 * VNB4 +
      DF125 * VNB5 + DF126 * VNB6 + DF127 * VNB7 + DF128 * VNB8 + DF129 * VNB9 + DF1210 * VNB10 +
      DF1211 * VNB11
  Food12C = DF121 * VWBsed + DF12p * VWBpart_DET + DF122 * VWB2 + DF123 * VWB3 + DF124 * VWB4
      + DF125 * VWB5 + DF126 * VWB6 + DF127 * VWB7 + DF128 * VWB8 + DF129 * VWB9 + DF1210 *
      VWB10 + DF1211 * VWB11
  Food12D = DF121 * VPBsed + DF12p * VPBpart_DET + DF122 * VPB2 + DF123 * VPB3 + DF124 * VPB4 +
      DF125 * VPB5 + DF126 * VPB6 + DF127 * VPB7 + DF128 * VPB8 + DF129 * VPB9 + DF1210 * VPB10 +
      DF1211 * VPB11
  Food12E = DF121 * CST + DF12p * CFL + DF122 * CB2 + DF123 * CB3 + DF124 * CB4 + DF125 * CB5 +
```

```
    DF126 * CB6 + DF127 * CB7 + DF128 * CB8_mf + DF129 * CB9 + DF1210 * CB10 + DF1211 * CB11
  GF12 = (((1 - eL12) * Food12A) + ((1 - eN12) * Food12B) + ((1 - eW12) * Food12C) + ((1 - eP12) * Food12D))
    * GD12
  VLG12 = ((1 - eL12) * Food12A) / (((1 - eL12) * Food12A) + ((1 - eN12) * Food12B) + ((1 - eW12) * Food12C)
    + ((1 - eP12) * Food12D))
  VNG12 = ((1 - eN12) * Food12B) / (((1 - eL12) * Food12A) + ((1 - eN12) * Food12B) + ((1 - eW12) * Food12C)
    + ((1 - eP12) * Food12D))
  VWG12 = ((1 - eW12) * Food12C) / (((1 - eL12) * Food12A) + ((1 - eN12) * Food12B) + ((1 - eW12) *
    Food12C) + ((1 - eP12) * Food12D))
  VPG12 = ((1 - eP12) * Food12D) / (((1 - eL12) * Food12A) + ((1 - eN12) * Food12B) + ((1 - eW12) * Food12C)
    + ((1 - eP12) * Food12D))
  ED12 = 1 / (EDA * KOW + EDB)
  KD12 = ED12 * GD12 / WB12
  EWW12 = 1 / (1.85 + (155 / KOW))
  K112 = EWW12 * GV12 / WB12
  KPW12 = (VLB12 * KOW) + (VNB12 * BETA * KOW) + VWB12
  K212 = K112 / KPW12
  Zorg12 = (VLB12 * Zlipid) + (VNB12 * BETA * Zlipid) + (VWB12 * Zwater)
  Zgut12 = VLG12 * Zlipid + VNG12 * BETA * Zlipid + VPG12 * GAMMA * Zlipid + VWG12 * Zwater
  KGB12 = Zgut12 / Zorg12
  KE12 = KGB12 / WB12 * ED12 * GF12
  CB12 = (CWB * K112 * (1 - FPW12) + CSD * K112 * FPW12 + KD12 * Food12E) / (K212 + KE12 + KG12 +
    KM12)
  dic.Add COMPARTMENT12, CB12

'MODEL CALCULATIONS FOR COMPARTMENT13 (BAS - bass)
  WB13 = Worksheets("Output").Cells(41, 16)
  VLB13 = Worksheets("Output").Cells(42, 16)
  VWB13 = Worksheets("Output").Cells(43, 16)
  VNB13 = Worksheets("Output").Cells(45, 16)
  VPB13 = 0
  WBL13 = WB13 * VLB13
  QW13 = 88.3 * WB13 ^ 0.6
  QL13 = QW13 * 0.01

  'Temperature dependent growth
  If TW < 17.5 Then
    KG13 = 0.000502 * WB13 ^ -0.2
  Else
    KG13 = 0.00251 * WB13 ^ -0.2
  End If

  GV13 = (1400 * (WB13 ^ 0.65)) / COX
  GD13 = 0.022 * WB13 ^ 0.85 * Exp(0.06 * TW)
  DF131 = Worksheets("Diet").Cells(14, 2)
  DF13p = Worksheets("Diet").Cells(14, 3)
  DF132 = Worksheets("Diet").Cells(14, 4)
  DF133 = Worksheets("Diet").Cells(14, 5)
  DF134 = Worksheets("Diet").Cells(14, 6)
```

```
DF135 = Worksheets("Diet").Cells(14, 7)
DF136 = Worksheets("Diet").Cells(14, 8)
DF137 = Worksheets("Diet").Cells(14, 9)
DF138 = Worksheets("Diet").Cells(14, 10)
DF139 = Worksheets("Diet").Cells(14, 11)
DF1310 = Worksheets("Diet").Cells(14, 12)
DF1311 = Worksheets("Diet").Cells(14, 13)
DF1312 = Worksheets("Diet").Cells(14, 14)
eL13 = Worksheets("Output").Cells(46, 16)
eP13 = Worksheets("Output").Cells(47, 16)
eN13 = Worksheets("Output").Cells(48, 16)
eW13 = Worksheets("Output").Cells(49, 16)
FPW13 = Worksheets("Output").Cells(50, 16)
Food13A = DF131 * VLBsed + DF13p * VLBpart_DET + DF132 * VLB2 + DF133 * VLB3 + DF134 * VLB4 +
    DF135 * VLB5 + DF136 * VLB6 + DF137 * VLB7 + DF138 * VLB8 + DF139 * VLB9 + DF1310 * VLB10 +
    DF1311 * VLB11 + DF1312 * VLB12
Food13B = DF131 * VNBsed + DF13p * VNBpart_DET + DF132 * VNB2 + DF133 * VNB3 + DF134 * VNB4 +
    DF135 * VNB5 + DF136 * VNB6 + DF137 * VNB7 + DF138 * VNB8 + DF139 * VNB9 + DF1310 * VNB10 +
    DF1311 * VNB11 + DF1312 * VNB12
Food13C = DF131 * VWBsed + DF13p * VWBpart_DET + DF132 * VWB2 + DF133 * VWB3 + DF134 * VWB4
    + DF135 * VWB5 + DF136 * VWB6 + DF137 * VWB7 + DF138 * VWB8 + DF139 * VWB9 + DF1310 *
    VWB10 + DF1311 * VWB11 + DF1312 * VWB12
Food13D = DF131 * VPBsed + DF13p * VPBpart_DET + DF132 * VPB2 + DF133 * VPB3 + DF134 * VPB4 +
    DF135 * VPB5 + DF136 * VPB6 + DF137 * VPB7 + DF138 * VPB8 + DF139 * VPB9 + DF1310 * VPB10 +
    DF1311 * VPB11 + DF1312 * VPB12
Food13E = DF131 * CST + DF13p * CFL + DF132 * CB2 + DF133 * CB3 + DF134 * CB4 + DF135 * CB5 +
    DF136 * CB6 + DF137 * CB7 + DF138 * CB8_mf + DF139 * CB9 + DF1310 * CB10 + DF1311 * CB11 +
    DF1312 * CB12
GF13 = (((1 - eL13) * Food13A) + ((1 - eN13) * Food13B) + ((1 - eW13) * Food13C) + ((1 - eP13) * Food13D))
    * GD13
VLG13 = ((1 - eL13) * Food13A) / (((1 - eL13) * Food13A) + ((1 - eN13) * Food13B) + ((1 - eW13) * Food13C)
    + ((1 - eP13) * Food13D))
VNG13 = ((1 - eN13) * Food13B) / (((1 - eL13) * Food13A) + ((1 - eN13) * Food13B) + ((1 - eW13) * Food13C)
    + ((1 - eP13) * Food13D))
VWG13 = ((1 - eW13) * Food13C) / (((1 - eL13) * Food13A) + ((1 - eN13) * Food13B) + ((1 - eW13) *
    Food13C) + ((1 - eP13) * Food13D))
VPG13 = ((1 - eP13) * Food13D) / (((1 - eL13) * Food13A) + ((1 - eN13) * Food13B) + ((1 - eW13) * Food13C)
    + ((1 - eP13) * Food13D))
ED13 = 1 / (EDA * KOW + EDB)
KD13 = ED13 * GD13 / WB13
EWW13 = 1 / (1.85 + (155 / KOW))
K113 = EWW13 * GV13 / WB13
KPW13 = (VLB13 * KOW) + (VNB13 * BETA * KOW) + VWB13
K213 = K113 / KPW13
Zorg13 = (VLB13 * Zlipid) + (VNB13 * BETA * Zlipid) + (VWB13 * Zwater)
Zgut13 = VLG13 * Zlipid + VNG13 * BETA * Zlipid + VPG13 * GAMMA * Zlipid + VWG13 * Zwater
KGB13 = Zgut13 / Zorg13
KE13 = KGB13 / WB13 * ED13 * GF13
CB13 = (CWB * K113 * (1 - FPW13) + CSD * K113 * FPW13 + KD13 * Food13E) / (K213 + KE13 + KG13 +
    KM13)
dic.Add COMPARTMENT13, CB13
```

```
'MODEL CALCULATIONS FOR COMPARTMENT14 (BC - blue crab)
   WB14 = Worksheets("BC").Cells(5, 5)
   VLB14 = Worksheets("BC").Cells(6, 5)
   VWB14 = Worksheets("BC").Cells(7, 5)
   VNB14 = Worksheets("BC").Cells(9, 5)
   VPB14 = 0
   WBL14 = WB14 * VLB14
   QW14 = 88.3 * WB14 ^ 0.6
   QL14 = QW14 * 0.01

   'Temperature dependent growth
   If TW < 17.5 Then
      KG14 = 0.000502 * WB14 ^ -0.2
   Else
      KG14 = 0.00251 * WB14 ^ -0.2
   End If

   GV14 = (1400 * (WB14 ^ 0.65)) / COX
   GD14 = 0.022 * WB14 ^ 0.85 * Exp(0.06 * TW)
   DF141 = Worksheets("BC").Cells(33, 2)
   DF14p = Worksheets("BC").Cells(33, 3)
   DF142 = Worksheets("BC").Cells(33, 4)
   DF143 = Worksheets("BC").Cells(33, 5)
   DF144 = Worksheets("BC").Cells(33, 6)
   DF145 = Worksheets("BC").Cells(33, 7)
   DF146 = Worksheets("BC").Cells(33, 8)
   DF147 = Worksheets("BC").Cells(33, 9)
   DF148 = Worksheets("BC").Cells(33, 10)
   DF149 = Worksheets("BC").Cells(33, 11)
   DF1410 = Worksheets("BC").Cells(33, 12)
   DF1411 = Worksheets("BC").Cells(33, 13)
   DF1412 = Worksheets("BC").Cells(33, 14)
   DF1413 = Worksheets("BC").Cells(33, 15)
   eL14 = Worksheets("BC").Cells(10, 5)
   eP14 = Worksheets("BC").Cells(11, 5)
   eN14 = Worksheets("BC").Cells(12, 5)
   eW14 = Worksheets("BC").Cells(13, 5)
   FPW14 = Worksheets("BC").Cells(14, 5)
   Food14A = DF141 * VLBsed + DF14p * VLBpart_DET + DF142 * VLB2 + DF143 * VLB3 + DF144 * VLB4 +
        DF145 * VLB5 + DF146 * VLB6 + DF147 * VLB7 + DF148 * VLB8 + DF149 * VLB9 + DF1410 * VLB10 +
        DF1411 * VLB11 + DF1412 * VLB12 + DF1413 * VLB13
   Food14B = DF141 * VNBsed + DF14p * VNBpart_DET + DF142 * VNB2 + DF143 * VNB3 + DF144 * VNB4 +
        DF145 * VNB5 + DF146 * VNB6 + DF147 * VNB7 + DF148 * VNB8 + DF149 * VNB9 + DF1410 * VNB10 +
        DF1411 * VNB11 + DF1412 * VNB12 + DF1413 * VNB13
   Food14C = DF141 * VWBsed + DF14p * VWBpart_DET + DF142 * VWB2 + DF143 * VWB3 + DF144 * VWB4
        + DF145 * VWB5 + DF146 * VWB6 + DF147 * VWB7 + DF148 * VWB8 + DF149 * VWB9 + DF1410 *
        VWB10 + DF1411 * VWB11 + DF1412 * VWB12 + DF1413 * VWB13
   Food14D = DF141 * VPBsed + DF14p * VPBpart_DET + DF142 * VPB2 + DF143 * VPB3 + DF144 * VPB4 +
        DF145 * VPB5 + DF146 * VPB6 + DF147 * VPB7 + DF148 * VPB8 + DF149 * VPB9 + DF1410 * VPB10 +
        DF1411 * VPB11 + DF1412 * VPB12 + DF1413 * VPB13
```

```
    Food14E = DF141 * CST + DF14p * CFL + DF142 * CB2 + DF143 * CB3 + DF144 * CB4 + DF145 * CB5 +
        DF146 * CB6 + DF147 * CB7 + DF148 * CB8_mf + DF149 * CB9 + DF1410 * CB10 + DF1411 * CB11 +
        DF1412 * CB12 + DF1413 * CB13
    GF14 = (((1 - eL14) * Food14A) + ((1 - eN14) * Food14B) + ((1 - eW14) * Food14C) + ((1 - eP14) * Food14D))
        * GD14
    VLG14 = ((1 - eL14) * Food14A) / (((1 - eL14) * Food14A) + ((1 - eN14) * Food14B) + ((1 - eW14) * Food14C)
        + ((1 - eP14) * Food14D))
    VNG14 = ((1 - eN14) * Food14B) / (((1 - eL14) * Food14A) + ((1 - eN14) * Food14B) + ((1 - eW14) * Food14C)
        + ((1 - eP14) * Food14D))
    VWG14 = ((1 - eW14) * Food14C) / (((1 - eL14) * Food14A) + ((1 - eN14) * Food14B) + ((1 - eW14) *
        Food14C) + ((1 - eP14) * Food14D))
    VPG14 = ((1 - eP14) * Food14D) / (((1 - eL14) * Food14A) + ((1 - eN14) * Food14B) + ((1 - eW14) * Food14C)
        + ((1 - eP14) * Food14D))
    ED14 = 1 / (EDA * KOW + EDB)
    KD14 = ED14 * GD14 / WB14
    EWW14 = 1 / (1.85 + (155 / KOW))
    K114 = EWW14 * GV14 / WB14
    KPW14 = (VLB14 * KOW) + (VNB14 * BETA * KOW) + VWB14
    K214 = K114 / KPW14
    Zorg14 = (VLB14 * Zlipid) + (VNB14 * BETA * Zlipid) + (VWB14 * Zwater)
    Zgut14 = VLG14 * Zlipid + VNG14 * BETA * Zlipid + VPG14 * GAMMA * Zlipid + VWG14 * Zwater
    KGB14 = Zgut14 / Zorg14
    KE14 = KGB14 / WB14 * ED14 * GF14
    CB14 = (CWB * K114 * (1 - FPW14) + CSD * K114 * FPW14 + KD14 * Food14E) / (K214 + KE14 + KG14 +
        KM14)
    dic.Add COMPARTMENT14, CB14

End Sub
```

# 4 References

Arnot JA, Gobas FAPC. 2004. A food web bioaccumulation model for organic chemicals in aquatic ecosystems. Environ Toxicol Chem 23:2343-2355.

EPA. 2014. Technical factsheet on: DIOXIN (2,3,7,8-TCDD). National Primary Drinking Water Regulations. US Environmental Protection Agency.

Gobas FAPC. 1993. A model for predicting the bioaccumulation of hydrophobic organic chemicals in aquatic food-webs: application to Lake Ontario. Ecol Model 69:1-17.

Morrison HA, Gobas FAPC, Lazar R, Haffner GD. 1996. Development and verification of a bioaccumulation model for organic contaminants in benthic invertebrates. Environ Sci Technol 30:3377-3384.

Morrison HA, Gobas FAPC, Lazar R, Whittle DM, Haffner GD. 1997. Development and verification of a benthic/pelagic food web bioaccumulation model for PCB congeners in Western Lake Erie. Environ Sci Technol 31:3267-73.

Shiu WY, Mackay D. 1986. A critical review of aqueous solubilities, vapor pressures, Henry's Law constants, and octanol-water partition coefficients of the polychlorinated biphenyls. J Phys Chem Ref Data 15:911-929.

Thomann RV, Connolly JP, Parkerton T. 1992. Modeling accumulation of organic chemicals in aquatic food webs. In: Gobas FAPC, McCorquodale JA, eds, Chemical dynamics in fresh water ecosystems. Lewis Publishers, Boca Raton, FL, pp 153-183.

Windward. [in prep]. Dissolved oxygen monitoring program data report for the Lower Passaic River study area: summer and fall 2012. Draft. Prepared for Cooperating Parties Group, Newark, NJ. Submitted to USEPA September 3, 2013. Lower Passaic River Restoration Project. Windward Environmental LLC, Seattle, WA.